

- 帮你总结应试问题
- 为你寻找解答思路

嵌入式系统设计师 案例导学

A Guidance Book for China Qualification Certificate
of Computer and Software Technology Proficiency

全国计算机技术与软件专业
技术资格（水平）考试辅导用书

郭春柱 编著



西安电子科技大学出版社
<http://www.xduph.com>

XDUP 213200

封面设计 张维涛

嵌入式系统设计师 案例导学

A Guidance Book for China Qualification Certificate
of Computer and Software Technology Proficiency

ISBN 978-7-5606-1840-1



9 787560 618401 >

定价：34.00元

全国计算机技术与软件专业技术资格(水平)考试辅导用书

嵌入式系统设计师 案例导学

郭春柱 编著

西安电子科技大学出版社

2007



内 容 简 介

本书根据 2005 年下半年实施的《全国计算机技术与软件专业技术资格(水平)考试——嵌入式系统设计师级考试大纲》精神,在深入研究历年计算机技术与软件专业技术资格(水平)考试下午试题的命题风格和试题结构的基础上,对考查的知识点进行了提炼,并对案例进行了分类。全书分为 9 章(第 0 章为考试指南),共编写了 25 个嵌入式系统设计案例。

本书以系统开发生命周期为基线,分别对嵌入式系统需求分析与体系结构设计、接口电路设计及其编程、嵌入式软件程序设计、嵌入式系统测试与维护、嵌入式 Linux 系统技术等领域的知识进行了案例讲解。每个章节包括学习要点、典型案例及真题链接三个部分。每一个案例均给出了详细的要点解析,其中不仅对试题的解题思路及步骤进行了讲解,而且对其考点及难点进行了剖析。第 6 章紧扣考试大纲,仿照 2006 年下半年考试真题的考核风格给出了两份模拟试卷,以增强考生学习的方向性和目的性。每份试卷均给出了相应的参考答案及要点解析。

本书可作为广大有志于通过嵌入式系统设计师考试人员的应试辅导用书,也可供各类高校、职高等院校的老师作为案例教学用书,还可作为各类计算机及网络工程等专业高年级学生获取嵌入式系统工程实践经验的参考用书。

图书在版编目(CIP)数据

嵌入式系统设计师案例导学/郭春柱编著. —西安:西安电子科技大学出版社,2007.7

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 978-7-5606-1840-1

I. 嵌… II. 郭… III. 微型计算机—系统设计—工程技术人员—资格考核—自学参考资料
IV. TP360.21

中国版本图书馆 CIP 数据核字(2007)第 080852 号

责任编辑 郑 东 张 梁 云立实

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 陕西华沐印刷科技有限责任公司

版 次 2007 年 7 月第 1 版 2007 年 7 月第 1 次印刷

开 本 787 毫米×1092 毫米 1/16 印 张 24

字 数 567 千字

印 数 1~4000 册

定 价 34.00 元

ISBN 978-7-5606-1840-1/TP·0957

XDUP 2132001-1

*** 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。



作者简介

郭春柱，嵌入式系统设计师，电子通信工程专业硕士。

曾获 2000 年度“中国建昊奖学金”。曾参加第五届全国大学生“挑战杯”课外学术作品竞赛、2000 年福建省大学生电子设计竞赛和 2001 年全国大学生电子设计竞赛，并获得了优异成绩。2005 年下半年，以上午 63 分、下午 59 分的成绩（全国前 50 名）通过网络工程师考试。2006 年下半年顺利通过系统分析师资格考试。

从写第一行单片机源代码到现在，从事政务网络规划、组建及嵌入式系统设计已经有 16 个年头，对“程序人生”也有了许多感触。很喜欢“程序人生”这个词，很确切。写程序与做人有很多相似的地方，都要诚实、正直。代码是一行行积累的，一点点的虚假都会给整个程序模块带来致命的错误，做人何尝不是以诚信为本！

对于嵌入式系统源代码编程、网络规划，本人不愿意一生只做这一件事情，但愿意干一生！

——郭春柱



前 言

► 本书编写目的

作为一本考试辅导用书，自然要做到“授之以渔”，而要撰写一本“授之以渔”的书籍，则并非易事。本书从内容上看，大致可以分为两部分。一方面，介绍了嵌入式系统设计师所必备的智力工具，即解题的知识点和答题过程等，文中涉及的概念较丰富，介绍的技术也较实用，它们是作者赠给读者的“鱼”。但是技术的发展一日千里，任何一门技术都不可避免有过时的一天。本书并没有受繁复技术的羁绊，而是高屋建瓴地抓住了隐藏在技术之后的原理、核心知识点以及发展脉络等“不变”的内容，而避开了“变化”迅速的技术细节。另一方面，作者在“渔”上下了很大的功夫，将一名嵌入式系统设计师所必须具备的基本知识及技能以系统开发生命周期为基线，分别对嵌入式系统需求分析与体系结构设计、接口电路设计及其编程、嵌入式软件程序设计、嵌入式系统测试与维护、嵌入式 Linux 系统技术等进行了案例讲解。每一个案例均给出了解答问题的详细逻辑推理过程。对每个嵌入式系统生命阶段需要掌握的技术进行扼要介绍，点到为止，并给出了如何获得相关资料和怎样掌握这些技术的方法。

全国计算机技术与软件专业技术资格(水平)考试(以下简称软件资格考试)作为一场由人事部和信息产业部直接领导下的权威性考试，决定了其考试范围的广度和深度都比较大，特别是许多考生在应对下午试题的复习过程中感觉无从下手，把握不住重要的知识点，其主要表现为备考知识点分布范围太广泛，缺少相关的应试阅读材料，手中掌握的材料重点不够突出以及备考思路无从建立等问题。

本书主要针对该级别下午试题范围广、难度大等考生反映的难点而编写，旨在“帮读者总结应试问题，为读者寻找解答思路”，为读者点亮备考路程中的导航灯，使读者更加明确努力的方向，在短时期内把握考试要领，从而减轻读者备考的负担，增强应试能力，从容应对考试。

► 本书特色

本书的最大特点是以考试大纲规定的考核点及能力层次为线索，按最新试题结构分章节进行编写。每个章节均列出可能出现的考核知识点，按考试题型编写对应的强化案例，以便读者能扎实、准确掌握本书所整理的知识内容。本书按最新试题结构编写了 25 个嵌入式系统试题案例并给出了详细的要点解析，其中不仅就试题进行解题思路及步骤的讲解和推理，而且对其考点及难点进行了剖析。其中，每一个案例都是融入作者在本门课程的教学经验并反复修改后定稿的，力求使读者的思路能从对庞杂的嵌入式系统工程知识点中得到升华。本书第 6 章紧扣考试大纲，仿照 2006 年下半年考试真题的考核风格给出了两份模拟试卷，以增强考生学习的方向性和目的性，每份试卷均给出了相应的参考答案及要点解析。相信本书对于准备参加资格(水平)考试的读者，复习下午试题案例的知识点、了解试

题形式和提高应试能力等均有较大帮助。

► 在线交流

为了更加有效地帮助读者冲刺下午试题,本书作者还在QQ群(27891542)上提供本书教学重点的PPT课件、相关章节的源程序、在线问题反馈和勘误表等内容。同时,为了进一步鼓励读者积极参与本书的勘误,笔者将对首个发现错误或积极提供建设性意见的读者,酌情赠送纪念品(例如本课程的考前冲刺试卷)。

由于笔者水平有限,且本书涉及的知识点较宽广,书中难免会存在一些错漏和不妥之处,恳请各位专家和读者在使用过程中多提批评性意见及建议,以利于本书质量的进一步改进和提高。笔者的E-mail地址是 guochunzhu@126.com。

► 致谢

在此,笔者向对本书出版给予关心、帮助与支持的每一位朋友表示衷心的感谢,没有他们的帮助,就不可能有本书的面世;感谢西安电子科技大学出版社云立实老师、郑东编辑和张梁编辑的悉心指导与热心支持;大学期间各位恩师的谆谆教诲使我受益匪浅,这些都使我感念不尽;感谢我周围的同事及朋友们,他们的支持与帮助使我能够更好地提高本书的质量;感谢父母亲的养育之恩和妻子谢秋玲的牺牲精神,他们生活上的照顾使我能够保持在学术的道路上不断进取,孜孜以求。

在本书出版之际,要特别感谢全国计算机技术与软件专业技术资格(水平)考试办公室的嵌入式系统设计师考试命题专家们,编者在本书中引用了该级别考试中的部分原题。同时,本书在编写过程中,参考了许多相关的书籍和因特网上的资料,在此对这些文献的作者也表示感谢。最后,衷心祝愿各位读者早日通过此项专业考试,成为一名合格的网络专业人才,也祝福祖国的计算机技术与软件事业蒸蒸日上。

郭春柱

2007年2月于福州

数字解密
PDG

目 录

第 0 章 考试指南	1
0.1 历年考题特点分析	1
0.2 考试大纲	4
0.2.1 考试说明	4
0.2.2 考试范围	5
0.2.3 题型举例	13
0.3 应试心得	14
第 1 章 嵌入式系统需求分析与体系结构设计	17
1.1 学习要点	17
1.1.1 考纲要求	17
1.1.2 学习建议	18
1.2 典型案例	19
1.2.1 案例 1 软件需求分析	19
1.2.2 案例 2 体系结构设计	23
1.2.3 案例 3 软硬件功能划分	30
1.2.4 案例 4 项目进度管理	36
1.2.5 案例 5 UML 建模技术	42
1.3 真题链接	51
1.3.1 2003 年系统分析员级下午试卷 I 试题 1	51
1.3.2 2005 年下半年系统分析员级下午试卷 I 试题 3	53
1.3.3 2006 年下半年系统分析员级下午试卷 I 试题 2	56
1.3.4 2006 年下半年嵌入式系统设计师下午试卷试题 1	58
第 2 章 接口电路设计及其编程	61
2.1 学习要点	61
2.1.1 考纲要求	61
2.1.2 学习建议	62
2.2 典型案例	63
2.2.1 案例 1 外接 Flash Memory 技术	63
2.2.2 案例 2 键盘接口电路	70
2.2.3 案例 3 LCD 接口电路设计及编程技术	81
2.2.4 案例 4 串口通信技术	89
2.2.5 案例 5 网络接口技术	97
2.3 真题链接	103
第 3 章 嵌入式软件程序设计	107
3.1 学习要点	107
3.1.1 考纲要求	107
3.1.2 学习建议	108
3.2 典型案例	109

3.2.1 案例1 任务实现技术	109
3.2.2 案例2 任务管理技术	115
3.2.3 案例3 程序设计技术	121
3.2.4 案例4 软件分析技术	127
3.2.5 案例5 文件系统管理	132
3.3 真题链接	137
3.3.1 2006年下半年嵌入式系统设计师下午试卷试题2	137
3.3.2 2006年下半年嵌入式系统设计师下午试卷试题3	140
3.3.3 2006年下半年嵌入式系统设计师下午试卷试题4	142
第4章 嵌入式系统测试与维护	144
4.1 学习要点	144
4.1.1 考纲要求	144
4.1.2 学习建议	144
4.2 典型案例	145
4.2.1 案例1 单元模块测试技术	145
4.2.2 案例2 覆盖测试技术	151
4.2.3 案例3 可移植性技术	158
4.2.4 案例4 可靠性分析技术	164
4.2.5 案例5 性能评价技术	169
第5章 嵌入式Linux系统技术	174
5.1 学习要点	174
5.1.1 考纲要求	174
5.1.2 学习建议	175
5.2 典型案例	175
5.2.1 案例1 inittab 配置文件	175
5.2.2 案例2 RPM 操作技术	182
5.2.3 案例3 内核移植与编译	187
5.2.4 案例4 Samba 配置技术	194
5.2.5 案例5 TCP/IP 网络配置	202
第6章 考点热身预测试题	208
6.1 模拟试题1	208
6.1.1 上午试题	208
6.1.2 下午试题	218
6.1.3 要点解析	229
6.1.4 参考答案	254
6.2 模拟试题2	256
6.2.1 上午试题	256
6.2.2 下午试题	265
6.2.3 要点解析	277
6.2.4 参考答案	299
第7章 2006年下半年嵌入式系统设计师试题	303
7.1 上午试题	303

7.1.1 试题描述	303
7.1.2 参考答案	311
7.1.3 要点解析	312
7.2 下午试题	328
第8章 上午试卷公共试题(中级)	329
8.1 2004年下半年真题链接	329
8.1.1 上午试题	329
8.1.2 参考答案	333
8.1.3 要点解析	333
8.2 2005年上半年真题链接	342
8.2.1 上午试题	342
8.2.2 参考答案	346
8.2.3 要点解析	346
8.3 2005年下半年真题链接	352
8.3.1 上午试题	352
8.3.2 参考答案	355
8.3.3 要点解析	355
8.4 2006年上半年真题链接	363
8.4.1 上午试题	363
8.4.2 参考答案	366
8.4.3 要点解析	367
8.5 2006年下半年真题链接	372
参考文献	373



第0章 考试指南

0.1 历年考题特点分析

考生在参加嵌入式系统设计师考试之前,除了按照考试大纲进行复习备考之外,还需要深入了解本级每次考试试卷的结构、题型和难度等情况。此外,还应该结合嵌入式系统技术的发展趋势来进行全面的考前准备,以提高自身的理论知识和基本技能。由于2006年下半年嵌入式系统设计师考试是本科目的首场考试,因此,分析此次的考题内容、题型的分值比例和题目难易程度等情况,将是考生备考过程中的领航灯。

嵌入式系统设计师考试试题分为上午试题和下午试题。其中,上午试题侧重基础知识,共有75道选择题,是每小题必做的客观题,每空1分,满分为75分。下午试题侧重技能运用,通常共有5道大题,每道大题15分,大部分试题是主观题,也有一小部分试题是客观选择题,应仔细按照试题要求简明扼要地解答问题。

从2006年下半年嵌入式系统设计师考试上午试卷的结构来看,此次考试遵循了计算机技术与软件专业技术资格(水平)考试(以下简称软件资格考试)模块化出题的风格,大体上可分为公共试题(即本级别其他科目也有的相同考题部分)和专业领域试题。其中,公共试题涵盖了计算机组成原理及体系结构、网络安全基础知识、标准化知识和知识产权保护知识、多媒体技术、软件工程、操作系统知识等方面的内容。这些试题涉及的内容比较广,所考查的知识点比较细,特别是近两年来,随着公共试题的考查风格、考试内容不断被考生所熟悉,使得这一部分试题的命题范围越来越广,所考查的知识点也越来越细,试题难度(主要体现在深度方面)也随之增大。这一部分试题的另一个特点是,计算题的比例约占该部分试题的40%~60%。2006年下半年上午考试中,试题1~试题22为公共试题部分,约占上午试卷的29.3%。其中,试题1~6、试题12~14、试题20和试题21属于计算题型,共有11小题,约占公共试题部分的50%。2006年下半年上午公共试题的考点分布统计表见表0-1。由表可知,计算机组成原理及体系结构、软件工程等部分所占的分值比例较高。建议考生一定要在这方面的知识点上多下功夫,努力做到熟能生巧,以便考试时能灵活变通,节约在这方面知识点的解题思考时间。

表0-1 2006年下半年上午公共试题的考点分布统计表

序号	考查内容	分值	题号	备注
1	计算机组成原理及体系结构	6分	1~6	
2	网络安全基础知识	3分	7~9	
3	标准化知识和知识产权保护知识	2分	10、11	
4	多媒体技术	3分	12~14	

续表

序号	考查内容	分值	题号	备注
5	软件工程	5分	15~19	
6	操作系统知识	3分	20~22	
7	计算机专业公共英语			
8	其他			

从2006年下半年嵌入式系统设计师考试上午试卷的专业领域试题(试题23~75)来看,其内容包括嵌入式系统的硬件基础、嵌入式微处理器体系结构、嵌入式系统的性能评价、嵌入式系统的存储体系、嵌入式系统输入/输出设备、嵌入式系统总线接口、嵌入式系统网络接口、电子电路设计基础、嵌入式软件基础、任务管理、存储管理、文件系统、嵌入式软件开发环境、嵌入式程序设计语言、嵌入式软件移植、系统分析基础知识、系统设计和系统实施知识等方面的内容。这些试题都注重基础理论知识的考查,所涉及的内容也比较广,但知识点考查的深度一般。其中还有不少考题是跨章节综合考核的,因此建议读者在备考过程中,要注意跨章节补充有关的综合应用知识以弥补自己知识上的盲点。表0-2给出了2006年下半年上午专业领域试题的考点分布统计情况。由表可知,“第2章 嵌入式微处理器与接口知识”、“第3章 嵌入式系统软件及操作系统知识”以及“计算机专业英语”等分别占有上午试卷13分、12分和10分,比例分别是17.3%、16%和13.3%。建议考生一定要在这方面的知识点上多下功夫,多阅读相关材料,多动笔做针对性的练习并从试题中多思考其出题的风格、出发点和形式等,以期直接获取应试经验,以便考试时能灵活变通,节约在这方面知识点的解题思考时间。注意,表0-2中2006年下半年上午专业领域试题没有考查的内容,不等于就是考试的非重点内容。例如,嵌入式系统中信息表示与运算基础、嵌入式系统电源和设备管理等都是嵌入式系统考试的重点内容,都有可能在今后试卷分值中占较大的比重。

表0-2 2006年下半年上午专业领域试题的考点分布统计表

章节	考查内容	分值	题号	备注
第1章 嵌入式系统基础知识	嵌入式系统的硬件基础	3分	23~25	
	嵌入式微处理器体系结构	3分	26~28	
	嵌入式系统的性能评价	1分	30	
	嵌入式系统的定义和组成			
	嵌入式系统中信息表示与运算基础			
第2章 嵌入式微处理器与接口知识	嵌入式系统的存储体系	2分	29、31	
	嵌入式系统输入/输出设备	1分	32	
	嵌入式系统总线接口	3分	33、34、36	
	嵌入式系统网络接口	2分	35、37	
	电子电路设计基础	4分	38~41	
	嵌入式微处理器的结构和类型			
	嵌入式系统电源			

续表

章 节	考 查 内 容	分 值	题 号	备 注
第 3 章 嵌入式系统软件及操作系统知识	嵌入式软件基础	2 分	42、43	
	任务管理	6 分	44~49	
	存储管理	2 分	50、51	
	文件系统	2 分	52、53	
	嵌入式操作系统概述			
	设备管理			
	其他	1 分	54	
第 4 章 嵌入式软件程序设计	嵌入式软件开发环境	1 分	55	
	嵌入式程序设计语言	5 分	56~60	
	嵌入式软件移植	1 分	61	
	嵌入式软件开发概述			
	嵌入式软件开发			
第 5 章 嵌入式系统开发与维护知识	系统分析基础知识	2 分	62、64	
	系统设计知识	1 分	65	
	系统实施知识	1 分	63	
	系统维护知识			
	系统开发过程及其项目管理			
第 6 章 嵌入式系统设计	嵌入式系统设计的特点			
	嵌入式系统的设计流程			
计算机专业英语		10 分	66~75	

表 0-3 给出了 2006 年下半年嵌入式系统考试下午试题的考点分布统计情况。由表中可以看出,此次的考核重点是“嵌入式软件程序设计”部分,共有三道大题,每题 15 分,此部分的分值约占下午试卷总分的 60%。此次试卷重点考查作为一名合格的嵌入式系统设计师应能根据实际应用环境的要求,进行系统的需求分析(试题 1 车载 GPS 终端系统 UML 建模技术)、组织和指导嵌入式系统开发人员实施嵌入式系统接口电路设计(试题 5 RS-232 异步串口接口电路设计)、编写和调试程序(试题 2 逻辑尺控制法 X86 汇编程序设计;试题 3 任务间的同步与互斥设计技术(C 语言);试题 4 实模式存储管理的内存地址空间布局)。注意,在表 0-3 中暂时没有出现的嵌入式系统项目管理技术、按照嵌入式系统总体设计规格进行软/硬件划分、系统开发规格说明书的编写、对嵌入式系统硬件设备和程序进行优化和集成测试、嵌入式 Linux 系统技术等内容均可能成为今后下午试题考核的重点内容,而且有可能在今后试卷分值中占较大的比重。在解答这类试题的过程中,考生除了熟悉有关理论、方法和配置步骤之外,还需要具有丰富的实践配置经验。如果没有机会实践,则需要多阅读相关案例,尽量从这些案例中间接获取经验。本书也将力求以发展的眼

光和实用的角度来预测、挖掘这些的相关考核点，以帮助读者在备考过程中分清主次、抓住重点，从而增强学习的目的性，减轻应试负担。

表 0-3 2006 年下半年嵌入式系统考试下午试题的考点分布统计表

序号	考查范围	考查内容	分值	题号	备注
1	需求分析与体系结构设计	车载 GPS 终端系统 UML 建模技术	15 分	试题 1	
2	接口电路设计及其编程	RS-232 异步串口接口电路设计	15 分	试题 5	
3	嵌入式软件程序设计	逻辑尺控制法 X86 汇编程序设计	15 分	试题 2	
		任务间的同步与互斥设计技术(C 语言)	15 分	试题 3	
		实模式存储管理的内存地址空间布局(C 语言)	15 分	试题 4	

由于嵌入式系统设计具有软硬件渗透、学科交叉的特点，从而决定了考试大纲涉及的内容多、知识面广，因此其中任何一方面的考点，都可能系统地写成一本书。本级别的出题风格和知识点的考核形式比较灵活，因此在今后的考试复习过程中，切不可死记硬背知识点，建议采用提纲挈要式地做笔记来准备考试。要对嵌入式系统所涉及的基本概念深入理解和学习，从嵌入式系统设计师的高度，结合平时嵌入式系统设计实践来体会考试大纲的考核点，并结合 2006 年下半年考试试题领会知识点，以便灵活掌握考试内容。受摩尔定律的支配，嵌入式系统的技术和模式也在不断地变化和发展着，这也同时要求读者不要仅囿于现有课本的知识范围，还要实时跟踪新技术的发展动态，关注其具有普遍性的问题。

0.2 考试大纲

0.2.1 考试说明

1. 考试要求：

- (1) 掌握科学基础知识；
- (2) 掌握嵌入式系统的硬件、软件知识；
- (3) 掌握嵌入式系统分析的方法；
- (4) 掌握嵌入式系统设计与开发的方法及步骤；
- (5) 掌握嵌入式系统实施的方法；
- (6) 掌握嵌入式系统运行维护知识；
- (7) 了解信息化基础知识、信息技术引用的基础知识；
- (8) 了解信息技术标准、安全，以及有关法律的基本知识；
- (9) 了解嵌入式技术发展趋势；
- (10) 正确阅读和理解计算机及嵌入式领域的英文资料。

2. 通过本考试的合格人员能根据项目管理和工程技术的实际要求，按照系统总体设计规格进行软、硬件划分，编写系统开发规格说明书等相应的文档；组织和指导嵌入式系统开发人员实施硬件电路、编写和调试程序，并对嵌入式系统硬件设备和程序进行优化和

集成测试, 开发出符合系统总体设计要求的高质量嵌入式系统; 具有工程师的实际工作能力和业务水平。

0.2.2 考试范围

一、考试科目 1: 嵌入式系统基础知识

1. 计算机科学基础

1.1 数制及转换

二进制、八进制、十进制和十六进制等常用数制及其相互转换

1.2 数据的表示

数的机内表示(原码、反码、补码、移码, 定点和浮点, 精度和溢出)

字符、汉字、声音、图像的编码方式

校验方法和校验码(奇偶校验码、海明校验码、循环校验码)

1.3 算术和逻辑运算

计算机中的二进制数运算方法

逻辑代数的基本运算和逻辑表达式的化简

1.4 计算机系统结构和重要部件的基本工作原理

CPU 和存储器的组成、性能、基本工作原理

常用 I/O 设备、通信设备的性能以及基本工作原理

I/O 接口的功能、类型和特点

虚拟存储器的基本工作原理, 多级存储体系

1.5 安全性、可靠性与系统性能评测基础知识

诊断与容错

系统可靠性分析评价

计算机系统性能评测方法

2. 嵌入式系统硬件知识

2.1 数字电路和逻辑电路基础

2.1.1 组合电路和时序电路

2.1.2 总线电路与电平转换电路

2.1.3 可编程逻辑器件

2.2 嵌入式微处理器基础

2.2.1 嵌入式微处理器体系结构

冯·诺伊曼结构与哈佛结构

CISC 与 RISC

流水线技术

信息存储的字节顺序(大端存储法和小端存储法)

2.2.2 嵌入式系统处理器的结构和类型

常用 8 位处理器的体系结构和类型

常用 16 位处理器的体系结构特点

常用 32 位处理器的体系结构特点

- 常用 DSP 处理器的体系结构特点
- 多核处理器的体系结构特点
- 2.2.3 异常
 - 同步异常(陷阱、故障、终止)
 - 异步异常(中断)
 - 可屏蔽中断、不可屏蔽中断
 - 中断优先级、中断嵌套
- 2.3 嵌入式系统的存储体系
 - 2.3.1 存储器系统
 - 存储器系统的层次结构
 - 高速缓存(Cache)
 - 内存管理单元(MMU)
 - 2.3.2 ROM 的种类与选型
 - 常见 ROM 的种类
 - PROM、EPROM、E2PROM 型 ROM 的典型特征和不同点
 - 2.3.3 Flash Memory 的种类与选型
 - Flash Memory 的种类
 - NOR 和 NAND 型 Flash Memory 的典型特征和不同点
 - 2.3.4 RAM 的种类与选型
 - 常见 RAM 的种类
 - SRAM、DRAM、DDRAM、NVRAM 的典型特征和不同点
 - 2.3.5 外存
 - 常见外存的种类
 - 磁盘、光盘、CF、SD 等的典型特征和不同点
- 2.4 嵌入式系统 I/O 接口
 - 2.4.1 定时器和计数器基本原理与结构
 - 2.4.2 GPIO、PWM 接口基本原理与结构
 - 2.4.3 A/D、D/A 接口基本原理与结构
 - 2.4.4 键盘、显示、触摸屏接口基本与结构
 - 2.4.5 嵌入式系统音频接口
- 2.5 嵌入系统通信及网络接口
 - PCI、USB、串口、红外、并口、SPI、IIC、PCMCIA 的基本原理与结构
 - 以太网、CAN、WLAN、蓝牙、IEEE 1394 的基本原理与结构
- 2.6 嵌入式系统电源分类及电源原理
- 2.7 电子电路设计
 - 2.7.1 电子电路设计基础知识
 - 电子电路设计原理
 - 电子电路设计方法及步骤
 - 电子电路设计中的可靠知识

2.7.2 PCB设计基础知识

PCB设计原理

PCB设计方法及步骤

多层PCB设计的注意事项及布线原则

PCB设计中的可靠性知识

2.7.3 电子电路测试基础知识

电子电路测试原理与方法

硬件抗干扰测试

3. 嵌入式系统软件知识

3.1 嵌入式软件基础知识

3.1.1 嵌入式软件的分类(系统软件、支撑软件、应用软件)

3.1.2 无操作系统支持的嵌入式软件体系结构(轮询、中断、前后台)

3.1.3 有操作系统支持的嵌入式软件体系结构

3.1.4 板极支持包基础知识(系统初始化、设备驱动程序)

3.1.5 嵌入式中间件(GUI、数据库)

3.2 嵌入式操作系统基础知识

3.2.1 嵌入式操作系统体系结构

单体结构、分层结构和微内核结构

3.2.2 任务管理

多道程序技术

进程、线程、任务的概念

任务的实现(任务的层次结构、任务控制块、任务的状态及状态转换、任务队列)

任务调度(调度算法的性能指标、可抢占调度、不可抢占调度、先来先服务、短作业优先算法、时间片轮转算法、优先级算法)

实时系统及任务调度(RMS、EDF算法)

任务间通信(共享内存、消息、管道、信号)

同步与互斥(竞争条件、临界区、互斥、信号量、死锁)

3.2.3 存储管理

Flat存储管理方式

分区存储管理(固定分区、可变分区)

地址重定位(逻辑地址、物理地址、地址映射)

页式存储管理

虚拟存储技术(程序局部性原理、虚拟页式存储管理、页面置换算法、工作集模型)

3.2.4 设备管理

设备无关性、I/O地址、I/O控制、中断处理、缓冲技术、假脱机技术

3.2.5 文件系统基础知识

文件和目录

- 文件的结构和组织
- 存取方法、存取控制
- 常见嵌入式文件系统(FAT、JFFS、YAFFS)
- 3.2.6 操作系统移植基础知识
- 3.3 嵌入式系统程序设计
 - 3.3.1 嵌入式软件开发基础知识
 - 3.3.2 嵌入式程序设计语言
 - 汇编、编译、解释系统的基础知识和基本工作原理
 - 汇编语言
 - 基于过程的语言(过程/函数、参数传递、全局变量、递归、动态内存分配、数据类型)
 - 面向对象的语言(对象、数据抽象、继承、多态、自动内存管理)
 - 各类程序设计语言的主要特点和适用情况
 - 3.3.3 嵌入式软件开发环境
 - 宿主机、目标机
 - 编辑器、编译器、链接器、调试器、模拟器
 - 常用嵌入式开发工具(编程器、硬件仿真器、逻辑分析仪、示波器)
 - 集成开发环境
 - 开发辅助工具
 - 3.3.4 嵌入式软件开发
 - 软件设计(模块结构设计、数据结构设计、内存布局、面向对象的分析与设计)
 - 嵌入式引导程序的设计、设备驱动程序设计、内核设计、网络程序设计、应用软件设计)
 - 编码(编程规范、代码审查)
 - 测试(测试环境、测试用例、测试方法、测试工具)
 - 下载和运行
 - 3.3.5 嵌入式应用软件移植
- 4. 嵌入式系统的开发与维护知识
 - 4.1 系统开发过程及其项目管理
 - 系统开发生命周期各阶段的目标和任务的划分方法
 - 系统开发项目基础知识及其常用管理工具使用方法
 - 主要的系统开发方法
 - 系统开发工具与环境知识
 - 4.2 系统分析基础知识
 - 系统分析的目的和任务
 - 系统分析方法
 - 系统规格说明书的编写方法
 - 4.3 系统设计知识

传统系统设计方法

软硬件协同设计方法

4.4 系统实施知识

系统架构设计

系统详细设计

系统调试技术

系统测试

4.5 系统维护知识

系统运行管理知识

系统维护知识

系统评价知识

5. 安全性知识

安全性基本概念

加密与解密机制

6. 标准化知识

标准化的概念

国际标准、国家标准、行业标准、企业标准基本知识

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知识

标准化机构

嵌入式系统相关标准

7. 信息化基础知识

信息化和信息系统基本概念

有关的法律、法规

8. 嵌入式技术发展趋势

9. 计算机专业英语

正确阅读和理解相关领域的英文资料

二、考试科目2：嵌入式系统设计应用技术

1. 嵌入式系统开发过程

1.1 系统需求分析方法与步骤

1.2 系统设计

系统硬件配置

系统功能组成分配

软硬件功能的分配

可行性验证及设计审查

系统规格

周期、成本及工作量估计

开发计划

1.3 软、硬件协同设计

1.4 硬件设计

- 1.5 软件设计
 - 软件结构
 - 设计评审
 - 软件详细设计
- 1.6 系统测试
 - 测试环境
 - 测试计划(内容、方法、标准、过程、检验)
 - 硬件测试
 - 软件测试(单元测试、集成测试)
 - 软硬件联合测试
 - 实施测试
- 1.7 系统评估
- 1.8 软件维护
- 2. 嵌入式系统硬件设计
 - 2.1 嵌入式系统硬件基本结构
 - 2.1.1 嵌入式微处理结构与应用
 - 2.1.2 异常及中断处理技术
 - 2.1.3 DMA 技术
 - 2.1.4 多处理系统
 - 多处理器系统特点
 - 多处理器系统构建技术
 - 2.1.5 总线架构
 - 应用系统中的总线配置
 - 2.1.6 内存种类及架构
 - 存储器系统接口设计
 - 2.1.7 数字电路和逻辑电路
 - 专用集成电路
 - 可编程逻辑控制器件
 - 2.2 输入/输出接口设计
 - 2.2.1 输入/输出接口
 - 接口信号电平转换
 - 接口驱动电路设计
 - 2.2.2 输入/输出接口应用技术
 - 外围设备
 - 串口通信
 - 并口通信
 - 模拟接口
 - 通信接口设备
 - 通信标准和协议



数据传输方式

2.3 外围设备接口应用技术

2.3.1 外围存储设备

存储卡、记忆棒、IC卡、MMC卡、SD卡
DVD、CD-R、CD-RW

2.3.2 外围输入/输出设备

键盘、鼠标、触摸屏
液晶板、LED、7段数码管、蜂鸣器

2.3.3 电源设计技术

2.4 可靠性与安全性设计技术

2.4.1 错误检测与隔离技术

2.4.2 冗余设计

2.4.3 系统恢复设计

2.4.4 诊断技术

2.4.5 常用安全标准

2.4.6 抗干扰设计

2.4.7 电磁兼容设计

2.4.8 系统加密

3. 嵌入式系统软件设计

3.1 嵌入式系统软件结构设计

3.2 嵌入式操作系统应用技术

3.2.1 时间管理

系统时间

时钟中断

3.2.2 内存管理

静态内存管理

动态内存管理

3.2.3 任务管理和任务间的通信

任务间的通信机制

信号量

邮箱

消息队列

3.2.4 异常处理

异常处理方法

中断优先级处理方法

系统调用

3.2.5 嵌入式文件系统应用技术

3.2.6 嵌入式系统图形用户接口(GUI)应用技术

3.2.7 嵌入式系统数据库应用技术

3.3 嵌入式软件设计技术

3.3.1 汇编语言设计

数据类型

汇编语言程序结构

汇编语言程序设计及优化

子程序调用

3.3.2 嵌入式 C 语言设计

ANSI-C 的数据类型

C 程序结构

C 语言程序设计及优化

程序的编译与链接

3.3.3 面向对象程序设计与开发

面向对象的分析与设计方法 UML

面向对象的编程语言

使用 C++ 进行嵌入式系统开发

使用 Java 进行嵌入式系统开发

3.4 系统级软件设计技术

嵌入式系统固件与系统初始化设计

设备驱动程序设计

硬件抽象层、板级支持包设计

嵌入式软件的移植技术

4. 嵌入式系统开发技术

4.1 系统开发环境

4.1.1 开发工具

文本编辑器

汇编、编译和连接程序

ICE 和 ICE 监控器

配置管理工具

逆工程工具

4.1.2 平台

操作系统

分布式开发环境

4.1.3 开发环境创建方法及评估

开发工作分析

开发环境的建立

维护、管理、使用开发环境的方法

开发环境的评测

4.2 实时系统的分析技术

4.2.1 实时系统的分析技术



- 结构化分析方法
- 面向对象分析方法
- 4.2.2 实时系统的设计技术
 - 结构化分析方法
 - 面向对象分析方法
- 4.3 硬件设计环境
 - 4.3.1 硬件描述语言
 - 硬件开发设计过程
 - 硬件描述语言的种类与特点
 - 4.3.2 仿真技术
 - 逻辑仿真方法
 - 逻辑仿真工具
 - 4.3.3 大规模集成电路系统的开发方法
 - ASIC 开发方法
 - FPGA 设计方法
 - IP(Intellectual Property)
- 4.4 协同设计
 - 软、硬件任务工作和协调
 - 设计评审
- 4.5 嵌入式系统低功耗设计技术
 - 低功耗系统工作机制
 - 低功耗系统模型结构
 - 低功耗的硬件设计技术
 - 低功耗的软件设计技术
- 4.6 分布式嵌入系统设计
 - 分布式系统设计原理
 - 分布式系统的通信技术
 - 分布式系统设计应用
- 5. 嵌入式系统应用
 - 5.1 嵌入式系统在控制领域中的应用
 - 5.2 嵌入式系统在手持设备中的应用
 - 5.3 嵌入式系统在模式识别中的应用

0.2.3 题型举例

一、选择题

若嵌入式系统中采用 I/O 地址统一编址模式,则访问内存单元和 I/O 设备是通过 ()来区分的。

A. 数据总线上输出的数据

B. 不同的地址代码

C. 不同的地址总路线

D. 不同的指令

二、问答题

在某个嵌入式操作系统中,任务的状态转换图(不完整)如图 0-1 所示,请阅读该图以及下列说明,然后回答问题 1 至问题 4,并将答案填入答题纸的对应栏内。

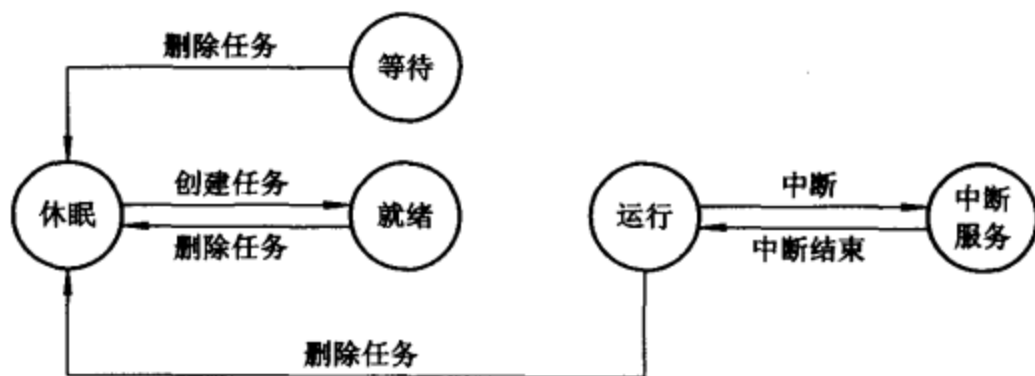


图 0-1 任务的状态转换图

【说明】

任务总共有五个状态,分别是休眠状态、就绪状态、运行状态、中断服务状态和等待状态。在任何时候,一个任务只会处于其中的某一个状态。

【问题 1】(3 分)

在单个 CPU 的系统中,处于运行状态的任务最多有多少个?

【问题 2】(3 分)

对于运行状态、就绪状态和等待状态之间的相互转换,图 0-1 中并没有画出来,请补充。用文字处理的形式来进行描述,格式形如“运行状态→中断服务状态”。

【问题 3】(6 分)

从运行状态可以直接变为等待状态吗?如果不能,为什么?如果能,什么时候会发生这种转换?举例说明。

【问题 4】(3 分)

从就绪状态可以直接变为等待状态吗?如果不能,为什么?如果能,什么时候会发生这种转换?举例说明。

0.3 应试心得

下面介绍一些作者在计算机技术与软件专业技术资格(水平)考试中的应试心得,与读者们一起分享。

(1) 摆正心态,做时间的主人。对待学习,一定要有主动的精神,无论什么事,只要是自己喜欢的并主动去做了的,你就一定会爆发出惊人的力量,从而把事情做得更好。有了主动学习的心态后,接下来非常重要的一件事是,在每天繁忙的学习工作中至少留出 1~2 个小时的复习时间。因为软件资格考试涉及的知识面较广、考查点深,你需要有足够的复习时间来夯实自己的专业基础知识。另外,最好能从周末抽出一部分时间,来回顾本周所复习的内容并对一些重要的知识点进行多角度的思考,预测命题人可能出题的形式。

(2) 厚积薄发，做知识的有心人。考试是一场智慧与毅力的较量，必须以深厚的专业知识作底蕴，用机敏的智慧沉着冷静地去分析、判断和取舍。一个人的精力和时间是有限的，在考前的复习过程中不能胡子眉毛一把抓，必须分清主次，抓住重点。在应试复习的过程中，对于诸如本书归纳总结的一些经典知识点要花力气多总结、多比较，找异同、抓规律，并努力做到熟能生巧，以便考试时能灵活变通，节约在这方面知识点的解题思考时间；同时也要通过网络、讲座、报纸、学术会议等途径及时了解一些技术热点和业界最新动态。

学习讲究的是勤奋和坚持，多劳多得，少劳少得，不劳不得。只观望而不学或学而不坚持的浮躁人群，永远成不了业界的先锋。

(3) 吃透真题，“它山之石，可以攻玉”。仔细分析一下历年考试的出题点，能拉近你与成功之间的距离。如果能把这一项工作做扎实，你的知识量应该就能够覆盖考试大纲中50%以上的知识点。在分析历年考题时，会发现嵌入式系统需求分析与体系结构设计、接口电路设计与编程、嵌入式系统软件设计、嵌入式系统测试与维护以及嵌入式Linux技术都可能成为每年考查的重要知识点。只要你真正掌握了这些知识点，通过嵌入式系统设计师的下午考试也不是件难事。

你可以利用本书第6章提供的两份模拟试卷严格按照考试时间进行实际训练，根据本书所提供的参考答案进行评分，从而了解自己的实际水平和差距。尤为重要的是，务必将所提供的要点解析的知识点一个个进行消化，吸收笔者在该知识点上所提供的应试经验，以期能够真正理解这些基础知识，并逐步做到举一反三，以不变应万变。

(4) 动手实验，在实践中锻炼提高。“纸上得来终觉浅，绝知此事要躬行”。如果说平时积累的各个知识点是一粒粒宝贵的珍珠的话，那么项目的实践经历则是将这些珍珠串联起来的一根红线。熟练掌握各种接口电路的设计、嵌入式实时操作系统软件的编写等都是一名嵌入式系统设计师开展日常技术支持、故障维护等工作的基本本领。在解答这类试题的过程中，考生除了熟悉有关理论、方法和配置步骤之外，还需要具有丰富的实践配置经验。另外，还需注意本书在相关程序命令的解释风格，在学习过程中可以依照该风格多下心思地解析每一条语句/命令，努力做到熟能生巧，以便考试时能灵活变通，节约在这方面知识点的解题思考时间。

(5) 掌握评卷专家心理，注意答题技巧。在嵌入式系统设计师下午试题的考试中，应注意把握评卷专家的心理状况。通常评卷专家不可能把你的论述一字一句地精读，要让他短时间内了解你的知识水平并认可你的能力，必须把握好主次关系，对于答案的组织一定要条理清晰，最好能够按主次关系分条进行陈述。此外，下午试题的分值大致按每个空格1分，或每个问题回答要点2分进行分布。这一点可以间接启示我们答题时可从每个问题的分值中大致估计出所需回答的要点数量。

(6) 摆正心态，轻装上阵。通过了嵌入式系统设计师考试并不代表就完全具备嵌入式系统设计所需的分析、设计和维护的能力。证书只是我们漫漫学习路途中一个阶段性的里程碑，通过学习来不断提高自己才是最终目的。考试前摆正心态也是件十分重要的工作，如果你在考试前一天晚上还看书到凌晨两三点，很可能第二天在考试时头脑昏沉，从而导致发挥失常。因此，应当以一颗平常心去应对考试，轻装上阵，这样才可能更好地发挥出自己的最佳状态，展示出应有的知识水平。

(7) 笑对成绩，雄关漫道从头越。对于考试成绩，笔者认为更应该保持一个良好的心态。其实，考试、成绩说到底它只是纸面上的东西，不能代表真正的能力与技术。证书只是一份“硬件”，更有价值的、更有用的是自己所掌握的技术和自身的能力等“软件”。考试的内容只是你进行工作所具备的必要理论基础，要想真正地成为一个嵌入式系统设计工程的高手，还必须多实践。这不是说考试就毫无价值，相反，理论知识是必须具备的，只有理论联系实际，才能做得更好。所以，大家无须太看重考试，把它看做是理论知识的一个凭证，努力修炼自己的内功才是根本。



第1章 嵌入式系统需求分析与体系结构设计

1.1 学习要点

1.1.1 考纲要求

根据考试大纲中相应的考核要求,在嵌入式系统的需求分析、规格说明、软硬件功能划分、系统体系结构定义和开发环境等方面,要求考生主要掌握以下方面的内容。

一、嵌入式系统开发过程

1. 系统需求分析方法与步骤

2. 系统设计

- (1) 系统硬件配置;
- (2) 系统功能组成分配;
- (3) 软、硬件功能的分配;
- (4) 可行性验证及设计审查;
- (5) 系统规格;
- (6) 周期、成本及工作量估计;
- (7) 开发计划。

3. 软、硬件协同设计

4. 硬件设计

5. 软件设计

- (1) 软件结构;
- (2) 设计评审;
- (3) 软件详细设计。

二、嵌入式系统开发技术

1. 系统开发环境

(1) 开发工具:

- 文本编辑器;
- 汇编、编译和连接程序;
- ICE 和 ICE 监控器;
- 配置管理工具;
- 逆工程工具。

(2) 平台：

- 操作系统；
- 分布式开发环境。

(3) 开发环境创建方法及评估：

- 开发工作分析；
- 开发环境的建立；
- 维护、管理、使用开发环境的方法；
- 开发环境的评测。

2. 实时系统的分析及设计

(1) 实时系统的分析技术：

- 结构化分析方法；
- 面向对象分析方法。

(2) 实时系统的设计技术：

- 结构化设计方法；
- 面向对象设计方法。

1.1.2 学习建议

熟练掌握嵌入式产品的需求分析、规格说明、软硬件功能划分和系统体系结构定义等技术，是一名嵌入式系统设计师开展嵌入式产品开发工作的基础。从2006年下半年嵌入式系统设计师考试下午试题的内容来看，在本章节知识点考核要求方面，主要是结合实际应用环境考查近阶段流行技术的一些基础知识，考试难度不大，即要求考生在该技术领域掌握知识的深度不高，大致处于“了解”的层次。解答此类试题，需要考生平时学习和积累，同时也要熟悉当前本领域的新技术和新方法的应用，最好能有一定的项目开发的应用实践经验。如果没有机会实践，则需要多阅读相关案例，尽量从这些案例中间接获取经验。本章也将力求以发展的眼光和实用的角度来预测、挖掘嵌入式系统产品定义方面的相关考核点，以增强考生学习相关知识点的目的性。

考试是一场智慧与毅力的较量，必须以深厚的专业知识作为底蕴，用机敏的智慧沉着冷静地去分析、判断、取舍。一个人的精力和时间是有限的，在考前的复习过程中必须分清主次，抓住重点。在应试条件下，对一些经典的知识点要花力气多总结、多比较、找异同、抓规律，并努力做到熟能生巧，以便考试时能灵活变通，节约在这方面知识点的解题思考时间。

通过本章学习，可获得以下理论知识点及工程实践经验：

- ① 嵌入式系统软件需求分析；
- ② 嵌入式系统体系结构设计；
- ③ 嵌入式系统软、硬件功能划分技术；
- ④ 嵌入式系统项目计划管理技术；
- ⑤ UML 建模技术。

1.2 典型案例

1.2.1 案例1 软件需求分析

一、案例描述

在设计一个嵌入式系统产品之前,必须先解决“做什么”的问题,即确定嵌入式系统产品的运行环境、功能及性能要求等设计任务和设计目标,并提炼出设计规格说明书,作为产品设计的指导和验收标准。

阅读以下关于某项嵌入式实时系统的软件需求的说明,根据要求回答问题1至问题4。(15分)

【说明】

某水电站根据安全监控的需要决定采用嵌入式实时控制系统进行实时多任务安全监测。该系统任务是按系统要求的固定的时间序列采集、处理和输出数据,以实现多台发电设备的综合控制。该实时控制系统经过对多个方案的对比、分析及讨论之后,硬件计划采用主频为133 MHz的PowerPC603e的32位微处理器、存储器、定时器、中断控制器、存储容量为2048KB的双口存储器、15路离散量接口、两路A/D、一路D/A数模转换器和四路422半双工串口接口(其中两路的频率为115 200 Hz,另两路的频率为38 400 Hz)组成。该实时控制系统的软件需求见表1-1。

表1-1 实时控制系统的软件需求表

任务简称	系统要求	处理任务的时间
10 ms 任务	以10 ms为周期从双口存储器中采集2048 KB的输入数据	约为2 ms
20 ms 任务	以20 ms为周期从两路422接口(115 200 Hz)中采集已预先进行格式定义的大小为128 B的数据。完成任务处理后,分别输出大小为16 B的控制命令	约为4 ms
60 ms 任务	以60 ms为周期从两路422接口(38 400 Hz)中采集已预先进行格式定义的大小为12 B的数据,从两路A/D接口采集28位数据。在完成处理(处理时间)任务后,分别输出大小为4 B的控制命令和一个28位D/A数据,输出10路离散量控制数据	约为3 ms
1 s 任务	作为系统的安全监控保障,系统要求在每1 s内对系统软硬件状态进行测试并完成系统工作的状态记录工作	5 ms

【问题1】(5分)

A公司承担了该嵌入式实时控制系统的软件开发任务,其开发小组根据用户的硬件环境及软件的需求,就软件的实施方案展开了激烈的讨论,讨论的首要问题就是本系统中软件的运行平台是采用商用嵌入式实时操作系统的开发方式还是在裸机上直接开发的方式。请用200字以内的文字简要说明采用这两种开发方式各有什么优缺点。请根据你的工程开

发经验,建议本项目采用哪种开发方式。

【问题 2】(3 分)

假设本项目的软件运行平台采用了简化的嵌入式 Linux 实时操作系统的开发方式,请在 250 字以内简要说明采用这种开发方式具有哪些优点。

【问题 3】(3 分)

对于实时操作系统,其主要任务是对事件进行实时的处理,虽然事件可能在无法预知的时刻到达,但是软件必须在事件随机发生时,在所限定的时间范围内做出响应(即系统的响应时间)。即使是系统处在尖峰负荷下也应如此,如果系统响应时间超时,就意味着致命的失败。A 公司开发小组成员郭工程师通过认真地对软件需求进行分析,给出了时间性能评估报告,判定在上述硬件平台上可以满足系统要求。

对于嵌入式实时操作系统中的时间性能评估,我们主要要考虑以下三项因素:

- ① 嵌入式实时操作系统的系统时间开销,一般不大于整个控制系统运行时间的(1)。
- ② 保证所有任务在规定的时间期限内完成(或进行可调度性评估)。
- ③ 任务的(2)和(3)。

【问题 4】(4 分)

根据本项目软件需求所提供的具体数据,请将以下(4)~(7)空缺处填写完整,并完成关于本实时控制系统的工作时序关系的叙述。

系统的最小时间节拍定义为(4),系统最大工作周期为(5),在每个 10 ms 起点,(6)任务首先运行,占(7)时间,完成后转入处理被终止的任务;如果这个时刻时间到达 20 ms 的起点,则启动 20 ms 任务运行,以此类推,确保在 1 s 周期内所有任务按规定的时间序列执行。

二、要点解析

【问题 1】(5 分)

这是一道要求读者进行嵌入式系统的软件运行平台选型的分析题。本题的分析思路如下:

① Donald Gillies 先生对实时操作系统的定义是,计算的正确性不仅取决于程序的逻辑正确性,也取决于结果产生的时间,如果系统的时间约束条件得不到满足,将会发生系统错误。

② 根据以上定义,在本项目中如果采用商用嵌入式实时操作系统进行开发,那么就要求该实时系统能够在限定的 10 ms、20 ms、60 ms 以及 1 s 等时间内分别提供表 1-1 中各周期任务的服务要求。

③ 通常根据实时系统对于响应时间的不同将其分为硬实时和软实时两种类型。其中,硬实时系统指系统要有确保的最坏情况下的服务时间,即对于事件的响应时间的截止期限无论如何都必须得到满足,如不能满足,将会带来严重的甚至灾难性的后果。例如航天领域中对宇宙飞船的控制就属于这种类型。而将除硬实时系统定义之外,其他的所有具有实时特性的系统称之为软实时系统。例如实时多媒体系统、液晶屏刷新时所允许的短暂延迟就属于这种类型。这种类型的实时系统只要一个任务就能够得到有确保的处理时间,到达系统的事件也能够在此截止期限到来之前得到处理,但违反截止期限并不会带来致命的

错误。

④ 如果本项目采用在裸机上直接开发的方式,并且关键部分采用汇编语言编写,那么可以保证该控制系统的实时性。

⑤ 该实时控制系统的软件处理工作将分别由 10 ms、20 ms 和 60 ms 等多种周期的处理任务组成,如果采用商用嵌入式实时操作系统作为本项目的开发基础,则可以保证系统的可靠性。

⑥ 综合以上分析、论证结果,建议本项目的软件运行平台采用商用嵌入式实时操作系统的开发方案。同时根据以上分析,归纳整理出符合题目字数要求的答案,即采用在裸机上直接开发的方式,并且关键部分采用汇编语言编写,可以保证该控制系统的实时性;由于本项目软件系统由多种周期的处理任务组成,采用商用嵌入式实时操作系统开发方式可以保证系统的可靠性。

另外需要说明的是,第③点所提到的硬实时系统和软实时系统在实现时的主要区别是选择不同的调度算法。对于软实时系统,选择基于优先级调度的算法足以满足软实时系统的需求,而且可以提供高速的响应和大的系统吞吐量;而对硬实时系统来说,需要使用的算法就应该是调度方式简单,反应速度快的实时调度算法。

【问题 2】(3 分)

这其实是一道要求读者说明采用嵌入式 Linux 实时操作系统开发方式理由的分析说理题。本题的解答思路如下:

① 嵌入式实时操作系统首先要完成的 4 个主要任务是进程管理、进程间通信与同步、内存管理和 I/O 资源管理。

② 实时操作系统的首要任务是调度一切可利用的资源完成实时控制任务,其次才着眼于提高计算机系统的使用效率。因此实时操作系统提供的任务调度功能,可以有效地管理本实时控制系统中 10 ms、20 ms 和 60 ms 等多种周期任务的调度。

③ 实时操作系统提供的事件、信号和任务间的通信机制,可以有效地解决本系统中资源共享时出现的互斥问题。

④ 另外,采用商品化的软件,不仅可以提高所开发软件的可靠性,还可简化软件开发成本,提高开发效率。

【问题 3】(3 分)

这是一道要求读者掌握对嵌入式实时操作系统进行时间性能评估需考虑哪些关键因素的分析题。本题的解答思路如下:

① 阅读题目后可知,题干所提供的信息强调的是实时操作系统的可确定性,即系统能对运行的最好和最坏情况做出精确的估计,这是实时操作系统的一个重要特点。

② 与实时操作系统相区别的是分时操作系统。对于分时操作系统,软件的执行在时间上的要求并不严格,时间上的延误或者时序上的错误一般不会造成灾难性的后果。

③ 通常,嵌入式实时操作系统的系统时间开销一般不大于整个控制系统运行时间的总开销的 20%。

④ 进行时间性能评估时还要保证所有任务在规定的时间内完成(或进行可调度性评估)。

⑤ 另外还要重点考虑任务的上下文切换时间和中断响应时间。其中,在嵌入式多任务

系统中,上下文切换是指 CPU 的控制权由运行任务转移到另外一个就绪任务时所发生的事件,当前运行任务转为就绪(或者挂起,或者删除)状态,另一个被选定的就绪任务成为当前任务。上下文切换包括保存当前任务的运行环境,恢复将要运行任务的运行环境。上下文的内容依赖于具体的 CPU。可见上下文切换时间是指执行多个任务时,系统发生任务切换,保存和恢复上下文的时间。而中断响应时间是指从发出中断请求到进入中断处理所用的时间。

【问题 4】(4 分)

这是一道要求读者根据实际环境分析嵌入式实时操作系统工作时序的试题。本题的解答思路如下:

① 根据问题要点,仔细阅读并找出题目所提供的关键信息。题干中关于软件需求的信息已将系统的任务按照执行的周期进行了分类,包括 10 ms 任务(即在 10 ms、20 ms 和 30 ms 等时刻,该任务需要被唤醒执行)、20 ms 任务(即在 20 ms、40 ms 和 60 ms 等时刻,该任务需要被唤醒执行)、60 ms 任务(即在 60 ms 和 120 ms 等时刻,该任务需要被唤醒执行)、1 s 任务(系统软、硬件测试任务)。

② 由于 10 ms 任务中处理任务的时间约为 2 ms,20 ms 任务中处理任务的时间约为 4 ms,60 ms 任务中处理任务的时间约为 3 ms,1 s 任务中处理任务的时间约为 5 ms,因此通过比较这些数据可知,系统最短的处理时间是 2 ms(10 ms 任务),最长的处理时间是 5 ms(1 s 任务)。

③ 通过以上分析,可以得到系统的最小时间节拍应定义为 10 ms,系统的主时间框架(即最大工作周期)为 1 s。

④ 在每个 10 ms 起点,10 ms 任务首先运行,占 2 ms 时间,完成后转入处理被终止的任务;如果这个转入时刻的时间到达 20 ms 的起点,则启动 20 ms 任务运行;如果这个转入时刻的时间到达 60 ms 的起点,则启动 60 ms 任务运行;以此类推,确保在 1 s 周期内所有任务按规定的时间序列执行。

图 1-1 示出了 10 ms、20 ms、60 ms 这三种周期任务的工作时序。注意,“处理任务的时间”和“任务处理时间”所表达的意思有些不一样,对于 1 s 任务系统只要求它每 1 s 内对系统软、硬件状态进行测试并完成系统工作的状态记录工作即可,并没有指定 5 ms 处理任务的时间具体在哪个时间段完成,可见这 5 ms 处理任务的时间的安排是灵活的,只要插入到系统有空闲的时间段即可。

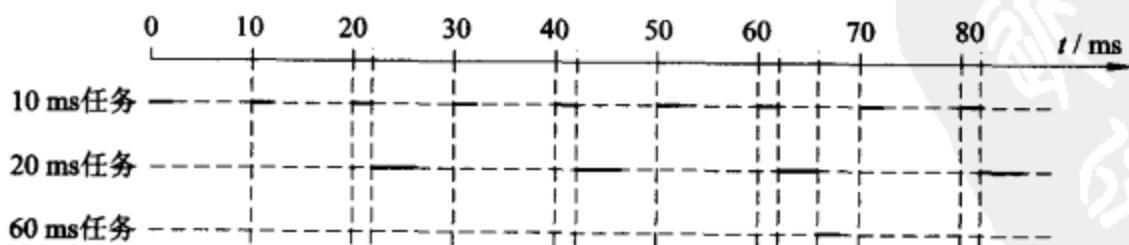


图 1-1 10 ms、20 ms 和 60 ms 等三种周期任务的工作时序示意图

三、参考答案

表 1-2 给出了本案例问题 1 至问题 4 的参考答案,供读者练习时参考,以便查缺补

漏。读者也可依照所给出的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

表 1-2 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (5 分)	采用在裸机上直接开发的方式,并且关键部分采用汇编语言编写,可以保证该控制系统的实时性(2 分) 采用商用嵌入式实时操作系统开发方式可以保证系统的可靠性(1 分) 由于本项目软件系统由多种周期的处理任务组成,为了保证系统的可靠性,建议采用商用嵌入式实时操作系统作为本项目的开发基础(2 分)	
【问题 2】 (3 分)	采用嵌入式实时操作系统开发方式的优点是: ① 实时操作系统提供的任务调度功能,可以有效地管理本系统多周期任务的调度(1 分); ② 操作系统提供的事件、信号和任务间的通信机制,可以有效地解决本系统中资源共享中的互斥问题(1 分); ③ 采用商品化的软件,可提高所开发软件的可靠性,还可简化软件开发成本,提高开发效率(1 分)	
【问题 3】 (3 分)	(1) 总开销的 20% (2) 上下文切换时间 (3) 中断响应时间(每空 1 分)	
【问题 4】 (4 分)	(4) 10 ms (5) 1 s (6) 10 ms (7) 2 ms(每空 1 分)	

1.2.2 案例 2 体系结构设计

一、案例描述

进行嵌入式系统产品的需求分析时,一般分功能性需求分析和非功能性需求分析两方面。功能性需求是系统的基本功能,如输入、输出信号,操作方式等;非功能性需求包括系统性能、成本、功耗、体积以及重量等因素。因此完成需求分析阶段的任务通常由两种过程来实现,分别是:① 先从客户那里收集系统的非形式描述;② 对客户需求进行提炼,得到系统的规格说明,这些规格说明里包含进行系统体系结构设计所需要的足够信息。

阅读以下关于某 GPS 移动地图开发项目的需求分析和体系结构设计的技术说明,根据要求回答问题 1 至问题 4。(15 分)

【说明】

A 公司是一个规模较大的汽车货运物流企业,物流网络覆盖东南沿海大部分省市和地区,各地办事处的位置相对比较分散,主要承接大客户长期的货物运载和配送服务,也面向中小客户提供货运搭配运载服务。A 公司为了能够查询车辆当前所在的位置并能够将车辆的信息显示在总公司新购建的地理信息系统(GIS)上,同时可随时根据需要调度车辆,对所管辖的所有车辆均安装了 GPS(全球卫星定位系统)相关设备。同时驾驶员可以通过一

种称为“GPS 移动地图”的手持设备，来显示他当前所处位置周围的地形，即地图的内容随着用户以及该设备所处位置的改变而改变，如图 1-2 所示。

假设准备对这种 GPS 移动地图进行嵌入式开发设计，而用户所提出的需求如下：

该手持设备主要针对高速公路上开车的驾驶员或类似用户，系统适应两种 GPS 系统的应用环境，能展示出可在标准地形图数据库中的得到的主要道路和其他陆地标志，所显示的地图有三种不同的尺寸精度，能在 400×600 像素分辨率的屏幕上平滑滚动所显示的地图。该设备的控制按钮数

不多于 3 个。按下按钮时，菜单系统应在屏幕上弹出，允许用户做出菜单选择。设备加电后在 1 s 内出现显示信息，驾驶员可以核查其位置并在 15 s 内显示当前地图，4 节 AA 电池至少可以连续开机运行 8 小时。

【问题 1】(4 分)

通常 GPS 移动地图设计项目的客户对嵌入式系统的理解是建立在他们想象的与系统之间交互的基础上的，因此从他们那里收集来的是非形式描述信息（即客户是使用他们自己的话而不是专业术语来表达其需求的）。这样，客户关于所需系统的描述和嵌入式系统设计师在体系结构设计时所需的信息之间存在较大的差距，所以必须将客户的描述转化为系统设计者的描述。

请从 GPS 移动地图用户所提出的需求中获取一组一致性的需求，然后从中整理成一份如表 1-3 所示的正式的规格说明表（书）。请将表 1-3 中(1)~(6)空缺处填写完整，以形成一份 GPS 移动地图设计项目的用户需求简表。

表 1-3 GPS 移动地图设计项目的用户需求简表

名称	(1)
目的	(2)
功能	(3)
输入	(4)
输出	(5)
性能	(6)
功耗	$\leq 120 \text{ mW}$
物理尺寸	$\leq 12 \text{ cm} \times 18 \text{ cm} \times 2 \text{ cm}$
重量	$\leq 400 \text{ g}$
生产成本	$\leq 680 \text{ 元}$

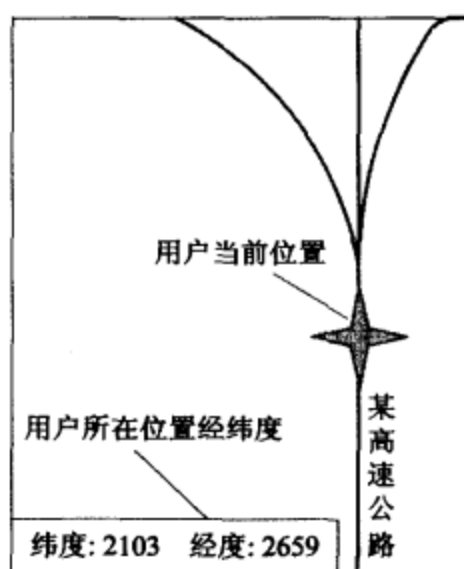


图 1-2 GPS 移动地图屏幕信息示意图

【问题 2】(4 分)

系统规格说明书反映了客户的需求并且是产品研发设计时必须明确遵循的要求，它起到客户和设计者之间的合同的作用。根据项目的设计需求，请说明该 GPS 移动地图设计项目的规格说明书应包含哪些构件。

【问题 3】(4 分)

规格说明书中只描述了系统“做什么”，而体系结构设计完成的是系统“如何做”的问题，即描述系统如何实现规格说明书中所定义的那些功能。体系结构是系统整体结构的一个计划，而后用于设计搭建整个体系结构的构件。

图 1-3 给出了 GPS 移动地图设计项目系统体系结构框图，它说明了一些主要操作及操作间的数据流。图 1-4 则给出了本设计项目的硬件框图，图中使用了两种存储器，一种是针对像素显示的帧缓冲器，另一种是 CPU 使用的通用程序/数据存储器。请将图 1-5 中 (7)~(10) 空缺处填写完整，以完成本项目的软件框图设计。

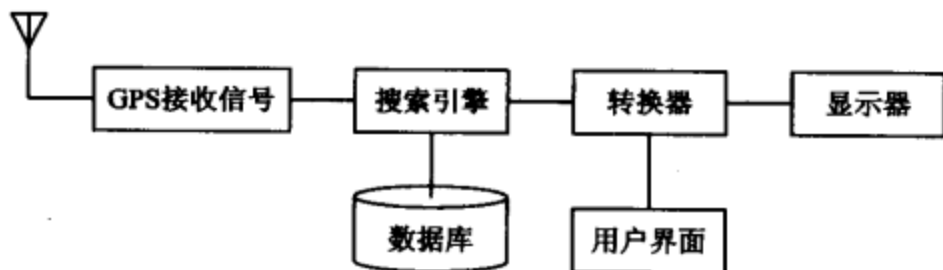


图 1-3 GPS 移动地图系统体系结构框图

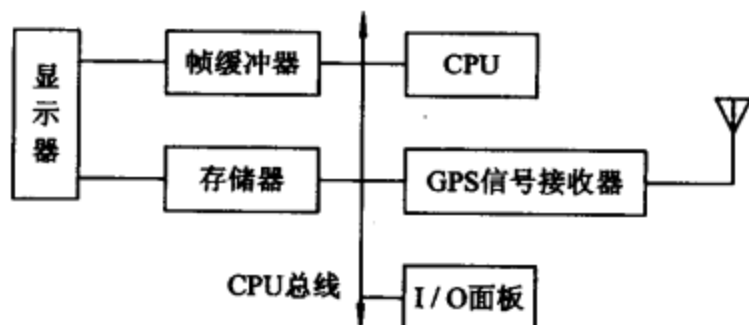


图 1-4 GPS 移动地图硬件框图

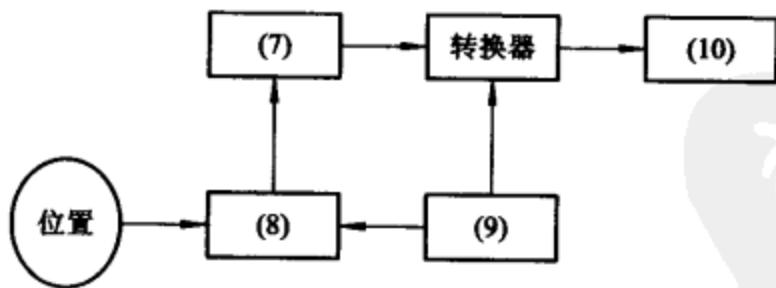


图 1-5 GPS 移动地图软件框图

【问题 4】(3 分)

将 GPS 移动地图的定位数据传回 A 公司信息中心 GIS 系统上，可以采用 GSM(全球移动通信)短信传输或 GPRS(通用无线分组业务)网络传输两种方式来实现。请根据你的工程经验，分析判断本项目采用哪种技术比较合适，并在 200 字以内简述你的选择理由。

二、要点解析

【问题 1】(4 分)

这是一道要求读者根据实际应用环境进行用户系统需求分析的试题。本题的解答思路如下：

① 系统的需求一般分功能性需求和非功能性需求两方面。功能性需求是系统的基本功能，例如表 1-3 中系统的名称、目的、功能、输入信号、输出信号以及操作方式等；非功能性需求包括系统的性能、功耗、物理尺寸(体积)、重量和生产成本等因素。

② 表 1-3 中“名称”栏(1)空缺处可填入“GPS 移动地图”，这一项比较简单，但却十分有用。给该项工程取一个名字可以使产品设计的目的更加明确，而且与客户或其他工程师讨论、沟通本工程项目时也比较方便。

③ “目的”栏中需要用简单的一两句话说出关于该系统所要满足的需求。(2)空缺处可从关键信息“该手持设备主要针对高速公路上开车的驾驶员或类似用户”中提炼，最后可填入类似于“为驾驶员提供的用户级移动地图”等内容。如果你不能用一两句话来描述你所设计的系统的目的，那么说明你还不是十分了解你所设计的项目。

④ “功能”是关于产品系统所要实现的任务的详细描述。通常可以从输入到输出进行分析来提取功能，一种常用的方法是通过做出类似以下的几个设问：

☆ 当系统接收到输入时，它执行哪些动作？

☆ 用户通过界面输入的数据如何对该功能产生影响？

☆ 不同功能之间是如何相互作用的？

从题目的“系统适应两种 GPS 系统的应用环境……所显示的地图有三种不同的尺寸精度”等信息中可以提取出相应的答案，即“使用两种接收器的 GPS 系统，三种用户可选的分辨率显示当前的经纬度”。

⑤ “输入”和“输出”栏所填的内容要考虑类似以下几个内容的大量的系统细节。

☆ 数据：是模拟信号还是数字信号或是机械输入？

☆ 输入/输出设备：按键数有几个？需要模/数转换器吗？是视频显示器还是 LCD 显示屏？

☆ 数据：是周期性到达的信号，还是用户按键的输入？每个数据元素包含多少位？

本题目所给的内容没有在这些方面做太多的描述，因此读者能提取出类似以下的答案即可：(4)空缺处“一个电源按钮，两个控制按钮”(此处所填写的内容受题目中“设备的控制按钮数不多于三个”这个信息限制)，(5)空缺处“LCD 显示，400×600 像素分辨率”(此处所填写的内容受限于题目中“手持设备”这个信息)。

⑥ 通常系统的性能是由执行一个用户级函数的大致时间和所需完成的一个特定操作的硬时限等因素组合而成的，而嵌入式系统的处理速度通常是该系统的实用性和最终成本的主要决定因素。因此在用户系统需求分析表中对性能的要求必须尽早地明确，这样才有助于利用这些性能要求随时检查所设计的嵌入式系统是否满足需求。

本试题中与(6)空缺处所填写内容的相关信息是“按下按钮时，菜单系统应在屏幕上弹出，允许用户对系统控制做出选择”。设备加电后在 1 s 内出现显示信息，驾驶员可以核查其位置并在 15 s 内显示当前地图，以及第⑤点的分析结果“LCD 显示”。考虑人眼的视觉暂

留效应及系统中其他因素(如软硬件工作时序、产品功耗及成本等),并借鉴我国市电频率 50 Hz 的常识,可得到类似“每隔 125 ms 动态刷新一次屏幕”等答案。

⑦ 表 1-3 中“功耗”栏,对于使用 4 节 AA 电池的本项目来说,电源电路的设计是非常重要的。顾客通常不会以瓦为单位来描述系统允许的功耗,因此需要根据用户需要进行估计,以做好用户与设计者之间语言的转换。

⑧ 对表 1-3 中物理尺寸和重量做出一定的规范有助于以后系统体系结构的设计。

⑨ 产品最终的成本也是用户需要分析表中一项主要的考虑因素。因为产品的成本包含两个主要部分:一是生产成本,包括购买构件以及组装它们的花费;二是不可再生的工程成本,包括人力成本以及设计系统的其他费用。“成本”栏将影响嵌入式系统硬件构件的选购等环节。

另外说明一点,对于一个大型嵌入式系统项目进行深入的需求分析需在以上要点解析①~③的基础上做出更长的需求文档。当做完一份需求分析报告(表)后,应该对它们的内部一致性进行检查,例如检查是否忘记了给某个功能指定输入或输出,是否考虑了系统运行的所有方式,是否把一些不切实际的要求规划到一个电池供电或低成本的设备中等。

【问题 2】(4 分)

这是一道要求读者掌握实际应用项目规格说明书设计的分析题。本题所涉及的知识如下:

① 系统规格说明书比用户需要表更精确一些,它起到客户和生产者之间的合同的作用。因此规格说明书必须小心编写,以便精确地反映客户的需求并且在设计时必须明确遵循的要求。

② 一份规格说明书的结构要明晰、有条理,以便其他人可以验证它是否符合系统需求及是否完全满足客户的期望。

③ 一份规格说明书所描述的内容不能有歧义,设计者一看就能知道什么是他们需要构造的。

④ 如果规格说明书的全局特征是错的或者是不完整的,那么由该规格说明建造的整个系统体系结构可能就不符合实现的要求。

⑤ 如果在某个特定的状况下的某些特性行为在规格说明中不明确,那么设计者可能会设计出部分错误的系统功能。

⑥ 做为一名嵌入式设计者要有足够的心理准备,工作过程中可能会碰到各种不同类型的由于不明确的规格说明而导致的方方面面的问题。

⑦ 该 GPS 移动地图设计项目的规格说明书主要包含从 GPS 卫星接收到的数据、地图数据、用户界面、系统所需执行的满足客户需求的操作以及系统运行时所需的后备动作(如操纵 GPS 接收机)等构件。

【问题 3】(4 分)

这是一道要求读者掌握嵌入式系统开发项目体系结构设计的分析题。本题的解答思路如下:

① 通俗地讲,一份系统体系结构报告告诉设计者的是需要什么样的构件。它以大构件为单位给出了系统的组成结构。

② GPS 移动地图系统体系结构框图(见图 1-3)中描述了如何实现规格说明书中所定

义的功能,例如系统需要通过搜索地形图数据库来绘制显示的结果。

③ 图 1-4 的系统硬件框图描述了系统使用了 CPU 器件,其周围还有存储器和 I/O 设备。图中还描述了系统使用了针对像素显示的帧缓冲器和通用程序/数据存储这两类存储器。图 1-4 还较详细地说明了 GPS 移动地图系统的硬件组成。

④ 图 1-5 软件框图的设计思路应源于图 1-3 系统体系结构框图,应保持两者的一致性。图 1-5 已给出了“转换器”的位置,根据这一信息并与图 1-3 对比,可以得出(7)空缺处应填入“数据库搜索”,与图 1-3 中“搜索引擎”相对应。

根据题干的“能在 400×600 像素分辨率的屏幕上平滑滚动所显示的地图……设备加电后在 1 s 内出现显示信息,驾驶员可以核查其位置并在 15 s 内显示当前地图”信息描述可知,该 GPS 移动地图通过屏幕平滑滚动来显示地图,并且允许用户核查显示信息中的具体位置信息。因此图 1-5 中“位置”信息应由图 1-3 中“用户界面”来获取,即(8)空缺处应填入“用户界面”。

根据图 1-5 中箭头方向,转换器完成的任务是将数据库搜索结果变换成显示器能接收的像素数据。由于题目要求“能在 400×600 像素分辨率的屏幕上平滑滚动所显示的地图”,因此需要从搜索数据库分离出图形绘制功能,这样有助于更平滑地更新屏幕,即数据库搜索与图形绘制功能应并行地完成各自的工作任务。(10)空缺处可填入类似“图形绘制”的答案。

另外,本 GPS 移动地图系统还需要 1 个计时器,用于控制何时读取用户界面上的按钮,并在屏幕上绘制数据等。(9)空缺处可填入用于产生各种定时信号的“时钟”。图 1-6 给出了完整的 GPS 移动地图软件结构框图。

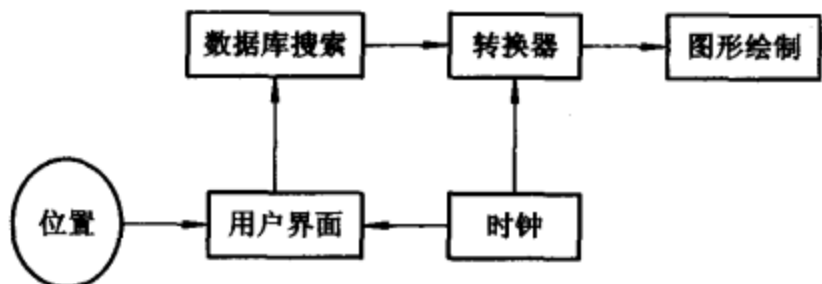


图 1-6 GPS 移动地图软件结构框图

在实际工程实施过程中还需要考虑更多的细节,例如软件框图中的单元

在硬件框图的什么地方执行,何时操作准时执行等。另外,体系结构设计阶段完成软硬件框图之后,还需要检查体系结构描述是否同时满足功能上和非功能上的用户需求,即体系结构设计框图中不仅要体现所需求的功能,而且必须符合成本、速度、功率和其他非功能上的约束。

如何知道硬件和软件体系结构是否符合速度、成本等方面的限制呢?通常需要有某种手段来估算框图中的构件,例如 GPS 移动地图系统中的搜索和绘制功能的特性。另外还需要借鉴以前的或他人的开发经验,最后能有类似系统的特定经验。有时还会考虑先构造一个简化的模型来帮助做出更精确的设计与估算。

【问题 4】(3 分)

这是一道要求读者比较 GSM 短信和 GPRS 技术特点并根据实际应用需求进行技术选型的分析题。本题的解答思路如下:

① GSM 短信系统是一种电路交换系统,适合传输少量的数据,数据可靠性好,系统简单且易于实现和维护。

② GPRS 是通用分组无线业务(General Packet Radio Service)的英文简称,它是在现

有的 GSM 系统上发展出来的一种新数据分组承载业务。它适合于传输大量的数据,传输数据的价格比较低。

③ GSM 系统与 GPRS 系统最根本的区别是, GSM 是一种电路交换系统,而 GPRS 是一种分组交换系统。

④ 目前,由于各地 GPRS 网络建设情况不一致,数据传输可靠性没有 GSM 短信好,并且本项目中 GPS 所传输的数据量不大,因此建议采用 GSM 短信传输方式。

根据以上分析,归纳整理出符合题目字数要求的答案,即 GSM 短信系统适合传输少量数据,数据可靠性好,系统简单且易于实现和维护。GPRS 适合于传输大量数据,传输数据的价格比较低。但由于各地 GPRS 网络建设情况不一致,数据传输可靠性没有 GSM 短信好,并且本项目中 GPS 所传输的数据量不大,因此建议采用 GSM 短信传输方式。

三、参考答案

表 1-4 给出了本案例问题 1 至问题 4 的参考答案,供读者练习时参考,以便查缺补漏。读者也可依照所给出的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

表 1-4 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (4 分)	(1) GPS 移动地图(0.5 分) (2) 为驾驶员提供的用户级移动地图(0.5 分) (3) 使用 2 种接收器的 GPS 系统(0.5 分); 3 种用户可选的分辨率显示当前的经纬度(0.5 分) (4) 1 个电源按钮, 2 个控制按钮(0.5 分) (5) 逆光 LCD 显示, 400×600 像素分辨率(0.5 分) (6) 每隔 125 ms 动态刷新一次屏幕(1 分)	
【问题 2】 (4 分)	① 从 GPS 卫星接收到的数据 ② 地图数据 ③ 用户界面 ④ 系统所需执行的满足客户需求的操作 ⑤ 系统运行时所需的后备动作(如操纵 GPS 接收机)等(答出 4 点即可, 每点 1 分, 最高得 4 分)	
【问题 3】 (4 分)	(7) 数据库搜索 (8) 用户界面 (9) 时钟 (10) 图形绘制(每空 1 分)	
【问题 4】 (3 分)	GSM 短信系统适合传输少量数据, 数据可靠性好, 系统简单且易于实现和维护(1 分) GPRS 适合于传输大量数据, 传输数据的价格比较低(1 分) 但由于各地 GPRS 网络建设情况不一致, 数据传输可靠性没有 GSM 短信好, 并且本项目中 GPS 所传输的数据量不大, 因此建议采用 GSM 短信传输方式(1 分)	

1.2.3 案例3 软硬件功能划分

一、案例描述^①

一个好的体系结构是一个嵌入式系统产品设计成功与否的关键。体系结构设计描述了系统如何实现所述的功能和非功能需求，包括对硬件、软件和执行装置的功能划分以及系统的软硬件划分、选型等。

阅读以下关于片上系统 SOC(System On Chip)设计中的软硬件功能划分的技术说明，根据要求回答问题 1 至问题 4。(15 分)

【说明】

某片上系统 SOC 由一个处理器和若干个运算部件构成，体系结构设计中的控制流程图如图 1-7 所示。该设计的约束条件是在时钟周期为 10 ns 的情况下，系统的延迟时间 ≤ 3850 ns。

可选用的处理器有 Pentium、PowerPC 和 6800 三类，每类处理器的相关指标如表 1-5 所示。

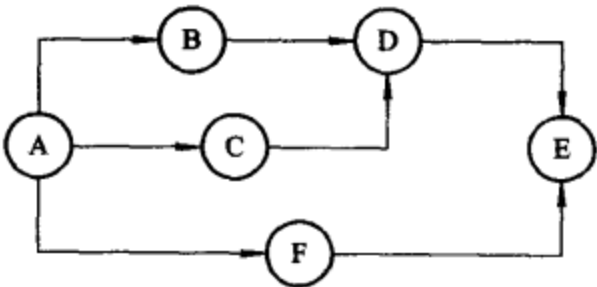


图 1-7 某 SOC 控制流程图

表 1-5 可选用的处理器

类 别	时 钟 周 期	价 格(元)
Pentium	10 ns	900
PowerPC	10 ns	750
6800	50 ns	600

表 1-6 给出了可供选择的硬件库，其中可选的器件有乘法器(Mul)、加法器(Add)以及比较运算器(Comp)。每个被选器件的名称、延时时间以及价格等如表 1-6 所示。

表 1-6 可选用的硬件库

类 别	名 称	延 时 时 间(ns)	门 数	价 格(元)
*	Mul1	30	80	90
*	Mul2	50	50	70
*	Mul3	70	45	50
+	Add1	30	40	35
+	Add2	42	35	30
>	Comp1	18	20	15
=	Comp2	14	15	10
<	Comp3	16	18	12

^① 本试题的题干描述参考了 2003 年系统分析师下午试题 I 试题 4 的相关内容。

经过测试,图 1-7 中流程图的各个节点在供选择的微处理器上的软件执行时间如表 1-7 所示。如果图 1-7 流程图中各节点选用硬件实现,则构成各个节点的器件(各节点内器件可以串联或并联工作)如表 1-8 所示。

表 1-7 节点的软件执行时间

节 点	处 理 器	执 行 时 间(ns)
A	Pentium	3100
	PowerPC	3800
	6800	6000
B	Pentium	1400
	PowerPC	2200
	6800	2800
C	Pentium	6800
	PowerPC	12 000
	6800	18 000
D	Pentium	900
	PowerPC	1000
	6800	1200
E	Pentium	12 000
	PowerPC	14 800
	6800	21 500
F	Pentium	850
	PowerPC	940
	6800	1060

表 1-8 各个节点器件表

节 点	构成的硬件
A	2 个 Mul1, 2 个 Mul3, 2 个 Add2, 2 个 Comp1
B	2 个 Mul2, 3 个 Add1, 1 个 Add2
C	2 个 Mul1, 1 个 Mul3, 2 个 Add2, 2 个 Comp2
D	2 个 Mul2, 2 个 Add1, 1 个 Add2, 2 个 Comp1
E	1 个 Mul1, 1 个 Mul2, 2 个 Add1, 2 个 Comp2
F	2 个 Mul2, 1 个 Mul3, 2 个 Add1, 1 个 Comp3

【问题 1】(4 分)

请根据所提供的信息,计算出图 1-7 控制流程图中节点 A~节点 E 用硬件实现方案

的成本及延时,将计算结果分别填入表 1-9 的(1)~(8)空缺处所对应的位置。

表 1-9 各节点用硬件实现方案的成本及延时

节点	成本(元)	延时(ns)
A	$2 \times 90 + 2 \times 50 + 2 \times 30 + 2 \times 15 = 370$	(1)
B	(2)	$2 \times 50 + 3 \times 30 + 1 \times 42 = 232$
C	(3)	(4)
D	(5)	(6)
E	(7)	(8)

【问题 2】(7 分)

如果在满足系统设计要求的前提下,要使最终产品成本最低,那么图 1-7 控制流程图中节点 A~节点 E 的哪些节点必须用硬件实现?哪些节点必须用软件实现?并简要说明理由。

【问题 3】(4 分)

如果要求最终产品的系统延迟时间 ≤ 3200 ns,那么软硬件的实现方案应如何改动才能使最终产品成本最低?

二、要点解析

【问题 1】(4 分)

这是一道要求读者根据实际应用进行硬件实现方案参数分析的试题。本题的解答思路如下:

① 对题目中表 1-9 已给出的提示信息进行深入的剖析,从中领悟出各节点用硬件实现方案中成本、延时的计算思路。

② 对于节点 A 成本的计算式子“ $2 \times 90 + 2 \times 50 + 2 \times 30 + 2 \times 15 = 370$ ”,分别在题目中找到式子左边各参数的具体含义。经过查找后,可以发现该式子的计算依据是表 1-8 各节点器件表以及表 1-6 硬件库中给出的每个被选器件的价格。为了便于理解,将式子做一次转换表达成以下式子,其中圆括号内容表示的是左边参数的具体含义:

$2(2 \text{ 个 Mul1}) \times 90(\text{每个 Mul1 的价格}) + 2(2 \text{ 个 Mul3}) \times 50(\text{每个 Mul3 的价格}) + 2(2 \text{ 个 Add2}) \times 30(\text{每个 Add2 的价格}) + 2(2 \text{ 个 Comp1}) \times 15(\text{每个 Comp1 的价格}) = 370(\text{节点 A 的总成本})$

③ 同理,对于节点 B 延时的计算式子“ $2 \times 50 + 3 \times 30 + 1 \times 42 = 232$ ”,分别在题目中表 1-8 各节点器件表以及表 1-6 硬件库中给出的每个被选器件的延时找到式子左边各参数的具体含义。为了便于理解,将式子转换表达成以下式子:

$2(2 \text{ 个 Mul2}) \times 50(\text{每个 Mul2 的延时}) + 3(3 \text{ 个 Add1}) \times 30(\text{每个 Add1 的延时}) + 1(1 \text{ 个 Add2}) \times 42(\text{每个 Add2 的延时}) = 232(\text{节点 B 的总延时})$

④ 根据以上分析思路,结合表 1-6 硬件库给出的每个被选器件的延时时间、价格和表 1-8 各节点器件表,分别计算出其他节点的硬件实现成本及延时。具体计算过程见表 1-10。

表 1-10 各节点用硬件实现方案的成本及延时

节点 A		节点 B	
构成的硬件	2 个 Mul1, 2 个 Mul3, 2 个 Add2, 2 个 Comp1	构成的硬件	2 个 Mul2, 3 个 Add1, 1 个 Add2
成本(元)	$2 \times 90 + 2 \times 50 + 2 \times 30 + 2 \times 15 = 370$	成本(元)	$2 \times 70 + 3 \times 35 + 1 \times 30 = 275$
延时(ns)	$2 \times 30 + 2 \times 70 + 2 \times 42 + 2 \times 15 = 320$	延时(ns)	$2 \times 50 + 3 \times 30 + 1 \times 42 = 232$
节点 C		节点 D	
构成的硬件	2 个 Mul1, 1 个 Mul3, 2 个 Add2, 2 个 Comp2	构成的硬件	2 个 Mul2, 2 个 Add1, 1 个 Add2, 2 个 Comp1
成本(元)	$2 \times 90 + 1 \times 50 + 2 \times 30 + 2 \times 10 = 310$	成本(元)	$2 \times 70 + 2 \times 35 + 1 \times 30 + 2 \times 15 = 270$
延时(ns)	$2 \times 30 + 1 \times 70 + 2 \times 42 + 2 \times 14 = 242$	延时(ns)	$2 \times 50 + 2 \times 30 + 1 \times 42 + 2 \times 18 = 238$
节点 E		节点 F	
构成的硬件	1 个 Mul1, 1 个 Mul2, 2 个 Add1, 2 个 Comp2	构成的硬件	2 个 Mul2, 1 个 Mul3, 2 个 Add1, 1 个 Comp3
成本(元)	$1 \times 90 + 1 \times 70 + 2 \times 35 + 2 \times 10 = 250$	成本(元)	$2 \times 70 + 1 \times 50 + 2 \times 35 + 1 \times 12 = 272$
延时(ns)	$1 \times 30 + 1 \times 50 + 2 \times 30 + 2 \times 14 = 168$	延时(ns)	$2 \times 50 + 1 \times 70 + 2 \times 30 + 1 \times 16 = 246$

【问题 2】(7 分)

这是一道要求读者根据实际应用进行各功能节点软硬件选择的分析题。本题的解答思路如下：

① 准确领会题目中关键信息“在满足系统设计要求的前提下，要使最终产品的成本最低”，其中“最低”间接说明本问题其实是求解图 1-7 控制流程图的最佳软件实现方案。

② 题干信息中已给出了系统设计时的惟一的约束条件“在时钟周期为 10 ns 的情况下，系统的延迟时间 ≤ 3850 ns”。由于表 1-5 中 6800 处理器的 1 个时钟周期为 50 ns (大于 10 ns)，因此 6800 处理器对于本项目的硬件实现是不可用，剔除该芯片，重点关注 Pentium 处理器和 PowerPC 处理器的延时及成本。将表 1-7 整理成表 1-11，其中“ > 3850 ”表示该节点的延时超过系统延迟时间的约束条件。

表 1-11 各节点用软件实现方案的成本及延时

节点	Pentium		PowerPC	
	延时/ns	成本/元	延时/ns	成本/元
A	3100	900	3800	750
B	1400		2200	
C	$6800 > 3850$		$12\ 000 > 3850$	
D	900		1000	
E	$10\ 020 > 3850$		$14\ 800 > 3850$	
F	850		940	

③ 由表 1-11 可知,如果单从软件延时角度考虑,那么节点 C、E 都必须使用硬件,因为节点 C 或 E 中任何一个不使用硬件来实现,整个系统流程的延迟时间将会超过 3850 ns。

④ 虽然试题中未对节点 F 提出是采用硬件实现还是采用软件来实现,但是“要使最终产品成本最低”却间接隐含着需要对节点 F 实现成本的考虑。由表 1-10 可知,节点 F 用硬件实现方案的成本是 272 元,处理延时是 246 ns。由表 1-11 可知,节点 F 在 Pentium 处理器上用软件实现方案的成本是 900 元,处理延时是 850 ns;而在 PowerPC 处理器上用软件实现方案的成本是 750 元,处理延时是 940 ns。

先从处理延时的角度考虑问题,由图 1-7 控制流程图可看出节点 F 与节点 A~节点 E 所组成的支路是并行工作的。结合表 1-10 和表 1-11 的数据可知,节点 F 不管是采用硬件实现还是采用软件实现均能满足系统延迟时间不超过 3850 ns 的要求,即整个系统的处理延时时间取决于节点 A~节点 E 所组成的支路的处理延时时间。从节点 F 实现成本的角度来看,在能满足系统印刷板制作、产品体积等要求的情况下,采用硬件实现方案的成本最低(272 元)。

⑤ 接下来考虑节点 A、B、D 是否使用软件来实现。现假设节点 A 用软件实现,节点 B、D 使用硬件来实现。那么整个流程的延迟时间为 $t_1 = 3100 \text{ ns}$ (节点 A, Pentium 处理器延时) + 242 ns(参见表 1-10, 节点 C 的硬件延时) + 238 ns(参见表 1-10, 节点 D 的硬件延时) + 168 ns(参见表 1-10, 节点 E 的硬件延时) = 3748 ns < 3850 ns(注意,节点 B 与节点 C 是并联执行的,而节点 C 的硬件延时 242 ns 大于节点 B 的硬件延时 232 ns,所以节点 D 要开始进行工作必须等待该并联环路中节点 C 的硬件延时执行完毕方可)。

⑥ 如果节点 B 用软件实现,节点 A、D 使用硬件来实现,那么整个系统流程的延迟时间为 $t_2 = 320 \text{ ns}$ (节点 A 的硬件延时) + 1400 ns(节点 B, Pentium 处理器延时) + 238 ns(节点 D 的硬件延时) + 168 ns(节点 E 的硬件延时) = 2126 ns < 3850 ns(注意,节点 B 与节点 C 是并联执行的,而节点 B 的软件延时 1400 ns 大于节点 C 的硬件延时 242 ns)。

⑦ 如果节点 D 用软件实现,节点 A、B 使用硬件来实现,那么整个系统流程的延迟时间为 $t_3 = 320 \text{ ns}$ (节点 A 的硬件延时) + 242 ns(节点 C 的硬件延时) + 900 ns(节点 D, Pentium 处理器延时) + 168 ns(节点 E 的硬件延时) = 1630 ns < 3850 ns(注意,节点 B 与节点 C 是并联执行的,而节点 C 的硬件延时 242 ns 大于节点 B 的硬件延时 232 ns)。

⑧ 通过以上分析可以得出:节点 A、B、D 只要其中一个节点可单独使用软件来实现,就能满足系统设计时的约束条件。接下来考虑这三个节点的组合使用软件来实现的情况。如果节点 A、B 同时用软件实现,而节点 D 用硬件来实现,那么查表 1-11 可知,对于 Pentium 处理器, 3100 ns (节点 A, Pentium 处理器软件延时) + 1400 ns (节点 B, Pentium 处理器软件延时) = 4500 ns > 3850 ns (延迟时间约束条件);对于 PowerPC 处理器, 3800 ns (节点 A 的处理器软件延时) + 2200 ns (节点 B 的处理器软件延时) = 6000 ns > 3850 ns (延迟时间约束条件)。综合以上两个分析结果,可以得出节点 A、B 不能同时用软件来实现。

⑨ 同理继续分析,如果节点 A、D 同时用软件来实现,而节点 B 用硬件来实现,那么查表 1-11 可知,对于 Pentium 处理器,整个系统流程的延迟时间为 3100 ns (节点 A 的处理器软件延时) + 242 ns (节点 C 的硬件延时,节点 B 与节点 C 是并联执行的,而节点 C 的硬件延时 242 ns 大于节点 B 的硬件延时 232 ns) + 900 ns (节点 D 的处理器软件延时)

+168 ns(节点 E 的硬件延时)=4410 ns > 3850 ns(延迟时间约束条件);对于 PowerPC 处理器, 3800 ns(节点 A 的处理器软件延时)+1000 ns(节点 D 的处理器软件延时)=4800 ns > 3850 ns(延迟时间约束条件)。综合以上两个分析结果, 可以得出节点 A、D 不能同时用软件来实现。

⑩ 如果节点 B 和 D 同时用软件实现, 而节点 A 用硬件来实现, 那么查表 1-11 可知, 对于 Pentium 处理器, 整个系统流程的延迟时间为 320 ns(节点 A 的硬件延时)+1400 ns(节点 B 的处理器软件延时, 节点 B 与节点 C 是并联执行的, 而节点 B 的处理器软件延时 1400 ns 大于节点 C 的硬件延时 242 ns)+900 ns(节点 D 处理器软件延时)+168 ns(节点 E 的硬件延时)=2788 ns 大于 3850 ns(延迟时间约束条件);对于 PowerPC 处理器, 320 ns(节点 A 的硬件延时)+2200 ns(节点 B 的处理器软件延时, 节点 B 与节点 C 是并联执行的, 而节点 B 的处理器软件延时 2200 ns 大于 C 的硬件延时 242 ns)+1000 ns(节点 D 处理器软件延时)+168 ns(节点 E 的硬件延时)=3688 ns < 3850 ns(延迟时间约束条件)。

⑪ 通过以上分析, 可以得出节点 B 和 D 能够同时用软件来实现。但对于 Pentium 处理器, 整个系统的成本 $M1=370$ (节点 A 的硬件成本)+900(节点 B 的处理器硬件成本)+310(节点 C 的硬件成本)+900(节点 D 的处理器硬件成本)+250(节点 E 的硬件成本)+272(节点 F 的硬件成本)=3002 元;而对于 PowerPC 处理器, 整个系统的成本 $M2=370$ (节点 A 的硬件成本)+750(节点 B 的处理器硬件成本)+310(节点 C 的硬件成本)+750(节点 D 的处理器硬件成本)+250(节点 E 的硬件成本)+272(节点 F 的硬件成本)=2702 元。由于 $M1>M2$, 题目又要求“在满足系统设计要求的前提下, 要使最终产品成本最低”, 因此要满足题目的要求必须满足节点 A、C、E、F 同时用硬件实现, 而节点 B 和 D 可同时用软件实现, 且需采用 PowerPC 处理器来使系统成本最低(2702 元)。

【问题 3】(4 分)

这道试题是对问题 2 的讨论内容进一步的延伸, 也是要求读者根据实际应用进行各功能节点软硬件选择的分析题。本题的解答思路可以借鉴问题 2 的分析过程。

① 由问题 2 分析过程中的第⑨点解答结果可知, 节点 A、C、E、F 可同时用硬件实现, 而节点 B 和 D 可同时用软件实现且均采用 Pentium 处理器时, 整个系统流程的延迟时间为 $2788\text{ ns} < 3200\text{ ns}$ (系统新的延迟时间约束条件), 此时整个系统的成本为 2730 元;而当节点 B 和 D 均采用 PowerPC 处理器时, 系统流程的延迟时间 $3688\text{ ns} > 3200\text{ ns}$ (系统新的延迟时间约束条件)。可见节点 B 和 D 不能同时采用 PowerPC 处理器。

② 现假设节点 B 采用 Pentium 处理器, 节点 D 采用 PowerPC 处理器, 此时整个系统流程的延迟时间为 320 (节点 A 的硬件延时)+ 1400 (节点 B 的 Pentium 处理器软件延时)+ 1000 (节点 D 的 PowerPC 处理器软件延时)+ 168 (节点 E 的硬件延时)= $2888\text{ ns} < 3200\text{ ns}$, 而整个系统的成本= $370+900+310+750+250+272=2852$ 元。

③ 假设 B 采用 PowerPC 处理器, D 采用 Pentium 处理器, 此时整个系统流程的延迟时间为 320 (节点 A 的硬件延时)+ 2200 (节点 B 的 PowerPC 处理器软件延时)+ 900 (节点 D 的 Pentium 处理器软件延时)+ 168 (节点 E 的硬件延时)= $3588\text{ ns} > 3200\text{ ns}$, 不能满足系统新的延迟时间约束条件, 因此不能采用这种实现方案。

④ 由于题目中要求“最终产品成本最低”, 因此综合以上分析结果, 可知节点 B 采用 Pentium 处理器, 节点 D 采用 PowerPC 处理器, 能满足“最终产品的系统延迟时间 $\leq 3200\text{ ns}$ ”。

且成本最低”这个条件。

三、参考答案

表 1-12 给出了本案例问题 1 至问题 3 的参考答案,供读者练习时参考,以便查缺补漏。读者也可依照所给出的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

表 1-12 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (4 分)	(1) 320 (2) 275 (3) 310 (4) 242 (5) 270 (6) 238 (7) 250 (8) 168 (每空 0.5 分)	
【问题 2】 (7 分)	节点 A、C、E、F 同时用硬件实现(3 分),而节点 B 和 D 可同时用软件实现(2 分)且需采用 PowerPC 处理器来使系统成本最低(2702 元)(1 分),此时整个系统流程的延迟时间为 3688 ns(1 分)	
【问题 3】 (4 分)	节点 A、C、E、F 都必须用硬件实现,且 B 和 D 不能同时采用 PowerPC 处理器(1 分) 节点 B 采用 Pentium 处理器,节点 D 采用 PowerPC 处理器(1 分),此时整个系统流程的延迟时间为 $320+1400+1000+168=2888\text{ ns}<3200\text{ ns}$ (1 分),系统成本 $=370+900+310+750+250+272=2852\text{ 元}$ (1 分)	

1.2.4 案例 4 项目进度管理

一、案例描述

阅读以下关于以快速原型模型开发嵌入式系统软件时的项目进度管理的说明,回答问题 1 至问题 5(15 分)。

【说明】

某嵌入式软件开发公司的谢工程师正在承接某项工控领域的嵌入式软件开发任务,在进行可行性研究时,需要估算完成项目的时间进度。由于此软件公司近年来已经为采用快速原型法开发软件提供了较好的规范、工具和辅助环境,谢工程师在开发此项目中准备采用图 1-8 所示的快速原型开发模型。

基于图 1-8 所示的快速原型开发模型,谢工程师在分析本项目的进度时,列出了以下三方面的任务。

(1) 软件开发主线任务,即采用快速原型法开发时每一个主线任务的工作步骤及其时间量估计,如表 1-13 所示。

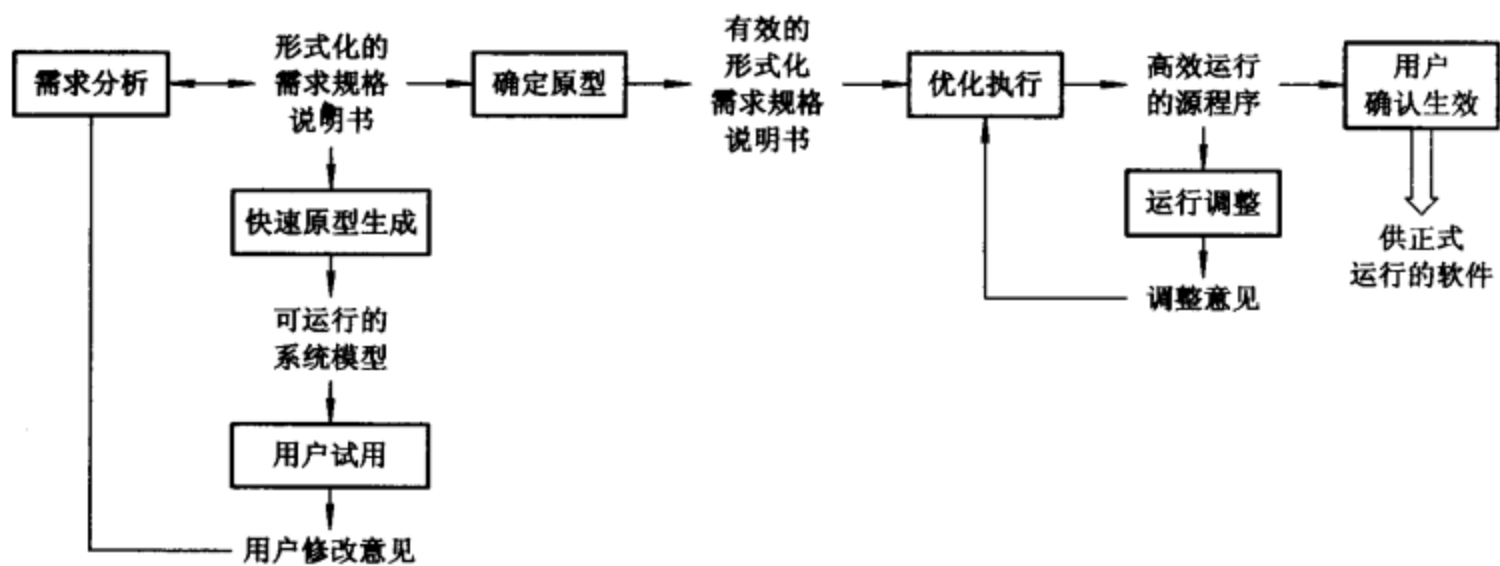


图 1-8 快速原型开发模型

表 1-13 软件开发主线任务表

序号	名 称	描 述
A	需求分析阶段	15 天，拟采用系统规格说明书流程图与结构化正文等作为主要描述工具，形成形式化的需求规格说明书
B1	快速原型生成阶段	第 1 次生成可运行系统原型约为 5 天，拟采用公司提供的系统开发规范语言转换描述后加以执行，生成可运行的原型
B2	用户试用原型阶段	第 1 次用户试用约为 10 天，通过试用得到用户修改意见
B3	修改需求分析阶段	第 1 次修改估计为 3 天，产生修改后的形式化的需求
C	确定原型阶段	6 天，通过与用户共同讨论分析，确定出有效的形式化需求规格说明书(用户须对此规格说明书表示满意)
D	优化执行阶段	5 天，即使用公司的辅助开发环境，把“有效的形式化需求规格说明书”转化成为可高效执行的源程序
E1	运行调整阶段	第 1 次估计需 8 天，通过运行获得需调整的意见
E2	维护性优化执行阶段	第 1 次估计需 2 天，产生调整后的可高效运行的源程序
F	用户确认生效阶段	估计为 6 天，最终获得可供正式投入运行的软件

表 1-13 中的 B1~B3 阶段还将要继续循环反复。根据经验，谢工程师估计最少还需要循环 3 次，其中快速原型生成分别需花费 3 天、2 天、1 天时间，用户试用分别需 4 天、2 天、1 天时间，修改需求分析分别需 2 天或 1 天时间(最后一次已无修改意见)。最多还可能需循环 7 次，前 3 次的循环各阶段所花费的时间同理想情况；但在后 4 次循环中，每次循环中快速原型生成、用户试用、修改需求分析都只需要 1 天时间就可以了。

同样，E1 和 E2 阶段也可能要继续反复循环。根据经验估计最少还需 1 次，其中需运行调整测试 2 天时间(无调整意见，不需要维护性优化执行)。最多可能还需 2 次，这时运行调整、维护性优化执行各需加上 2 天时间。

(2) 文档生成任务。该嵌入式软件开发公司十分强调加强对快速原型法开发软件时的

文档工作。谢工程师认为需求分析文档、原型及用户修改意见文档、有效需求分析文档、高效执行源程序文档、调整意见文档、用户生效意见及正式运行系统的文档等都应在表 1-13 中的 A~F 阶段内相应同时完成。同时还需要考虑 G. 用户使用说明书约需 12 天完成, H. 用户操作与维护说明书约需 12 天完成。

(3) 用户培训任务。在开发的过程中, 必须随时加强对用户的若干环节的培训, 如表 1-14 所示。

表 1-14 用户培训任务表

序号	任务描述	所需时间
I	在需求分析阶段开始 12 天后, 才能启动下一阶段的用户试用培训	
J	用户试用培训	约 10 天
K	用户理解需求规范、系统的各类特征及确定原型等培训	共 28 天
L	用户各类操作和使用方面的培训	约 14 天
M	用户常规维护等培训	约 10 天

根据以上(1)~(3)这三类任务的时序关系, 可以大致获得如图 1-9 所示的任务进度图。

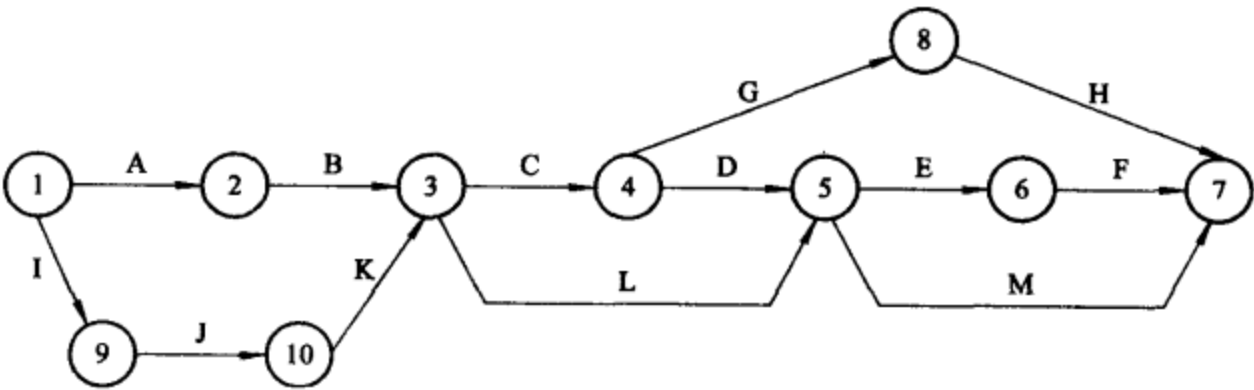


图 1-9 任务进度图

【问题 1】(3 分)

请在 100 字以内说明该项工控领域的嵌入式软件开发任务采用快速原型开发方法的优缺点。

【问题 2】(5 分)

根据试题的描述信息, 分析在最理想的情况下, 需要多少天才能完成此工控领域的嵌入式软件开发任务? 如果按保守的估计, 则需要多少天才可完成此开发任务? (请列出简要的计算过程)

【问题 3】(2 分)

请指出图 1-9 可能存在的关键路径是什么? (请用英文字母序号列出)

【问题 4】(2 分)

在最理想和保守的估计中加速开发进度要着重抓住的共同环节有哪些? 请在 50 字以内说明。

【问题 5】(3 分)

项目管理就是以项目为对象的系统管理方法, 通过一个临时性的专门的柔性组织, 对

项目进行高效率的计划、组织、指导和控制,以实现项目全过程的动态管理和项目目标的综合协调与优化。除了本题涉及到的项目进度管理之外,对于嵌入式软件项目的开发还需关注哪些方面的管理?

二、要点解析

【问题 1】(3 分)

这是一道要求读者掌握快速原型开发方法在嵌入式系统软件开发中应用的简答题。本题的解答思路是:题干的图 1-8 描述了快速原型模型的开发过程,即第一步是建造一个快速原型,实现客户或未来的用户与系统的交互,用户或客户对原型进行评价,进一步细化待开发软件的需求,通过逐步调整原型使其满足客户的要求,开发人员可以确定客户的真正需求是什么;第二步则在第一步的基础上开发客户满意的软件产品。可见,快速原型方法关注于满足客户需求,可以克服基于文档驱动的瀑布模型的缺点,减少由于软件需求不明确带来的开发风险,但也可能导致系统设计差、效率低、难以维护等问题。表 1-15 罗列出了几种常见的系统开发模型的优缺点。

表 1-15 几种常见的系统开发模型的优缺点

模 型	优 点	缺 点
瀑布模型	文档驱动	系统可能不满足客户的需求
快速原型模型	关注满足客户需求	可能导致系统设计差、效率低、难以维护
增量模型	开发早期反馈及时,易于维护	需要开放式体系结构,可能会设计差、效率低
螺旋模型	风险驱动	风险分析人员需要有经验且经过充分训练
喷泉模型	对象驱动,使开发过程具有迭代性和无间隙性	

【问题 2】(5 分)

这是一道要求读者掌握 PERT 图在嵌入式系统软件开发中应用的计算题。本题的解答关键是熟练查找出 PERT 图中的关键路径。本题的分析思路如下:

① 项目评估与评审技术(PERT)图是一种常用的安排软件开发进度,制定软件开发计划的项目管理工具,它采用网络图来描述一个项目中任务与任务之间依赖关系,是一种图形化的网络模型。

② 与普通的 PERT 图相比,本试题因为涉及采用快速原型模型开发该工控嵌入式软件,所以存在循环和迭代的过程。根据题干信息的描述以及图 1-9 中所给出的任务间的流程关系可知,本题存在着两个循环过程,分别是步骤 B1~B3 的循环和步骤 E1~E2 的循环。

③ 根据图 1-10 的示意,将 B1、B2、B3 的时间估计值之和作为 B 阶段的总时间,把 E1、E2 的时间估计值之和作为 E 阶段的总时间。

④ 在理想情况下,根据题干关键信息“B1~B3 阶段……最少还需要循环 3 次,其中快速原型生成分别需 3 天、2 天和 1 天,用户试用分别需 4 天、2 天和 1 天,修改需求分析分别需 2 天或 1 天”,结合表 1-13 中第 1 次 B1~B3 阶段所花费的时间可求解出 B 阶段的总时间 T_B 。

$$\begin{aligned}
 T_B &= \sum_1^4 (T_{B1} + T_{B2} + T_{B3}) \\
 &= (5 + 10 + 3) + (3 + 4 + 2) + (2 + 2 + 1) + (1 + 1 + 0) \\
 &= 34 \text{ 天}
 \end{aligned}$$

以上式子中，“0”表示最后一次已无修改意见的修改需求分析阶段所花费的时间。

⑤ 同理，在理想情况下，根据题干关键信息“E1 和 E2 阶段也可能要继续反复循环，根据经验估计最少还需 1 次，其中需运行调整测试 2 天（无调整意见，不需要维护性优化执行）”，结合表 1-13 中第 1 次 E1、E2 阶段所花费的时间可求解出 E 阶段的总时间 T_E 。

$$T_E = \sum_1^2 (T_{E1} + T_{E2}) = (8 + 2) + (2 + 0) = 12 \text{ 天}$$

⑥ 通常将关键路径定义为 PERT 图中最长的路径，也可以说是完成项目的最短的可能时间。如果关键路径上的任何任务停滞下来，则整个项目也将会停滞下来。

根据题干给出各个阶段所花费的时间以及 T_B 、 T_E 计算结果，得到理想情况下该工控嵌入式软件的双代号任务进度图，如图 1-10 所示。图 1-10 中存在 ABCDEF、ABCDM、ABCGH、ABLEF、ABLM、IJKDEF、IJKCDM、IJKCGH、IJKLEF、IJKLM 等 10 条路径。假设图 1-10 中箭头上的时间量是确定性的估计值，那么路径 ABLEF 需花费 $15 + 34 + 14 + 12 + 6 = 81$ 天，而路径 IJKLEF 需花费 $12 + 10 + 28 + 14 + 12 + 6 = 82$ 天。根据关键路径的定义可知，路径 IJKLEF 是图 1-10 中最长的路径，是理想情况下该软件开发任务进度的关键路径。

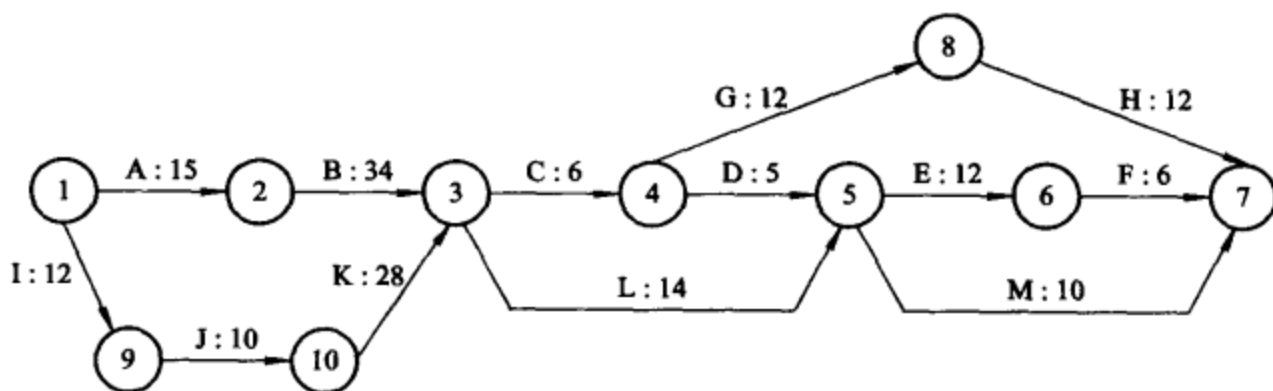


图 1-10 理想情况下任务进度图

由以上分析过程可知，在最理想的情况下，需要 82 天才能完成此工控领域的嵌入式软件开发任务。

⑦ 在保守情况下，根据题干关键信息“B1~B3 阶段……最多还可能需循环 7 次，前 3 次的循环各阶段所花费的时间同理想情况，但在后 4 次循环中，每次循环中快速原型生成、用户试用、修改需求分析都只需要 1 天就可以了”可求解出此时 B 阶段的总时间 T_B 。

$$\begin{aligned}
 T_B &= \sum_1^8 (T_{B1} + T_{B2} + T_{B3}) \\
 &= (5 + 10 + 3) + (3 + 4 + 2) + (2 + 2 + 1) + (1 + 1 + 0) + 4 \times (1 + 1 + 1) \\
 &= 34 + 12 = 46 \text{ 天}
 \end{aligned}$$

由题干关键信息“E1 和 E2 阶段……最多可能还需 2 次，这时运行调整、维护性优化执行各需加上 2 天时间”可求解出保守情况下 E 阶段的总时间 T_E 。

$$T_E = \sum_1^3 (T_{E1} + T_{E2}) = (8 + 2) + (2 + 2) + (2 + 0) = 16 \text{ 天}$$

⑧ 根据题干给出各个阶段所花费的时间以及保守情况下 T_B 、 T_E 计算结果, 得到保守情况下该工控嵌入式软件开发任务的双代号任务进度图, 如图 1-11 所示。

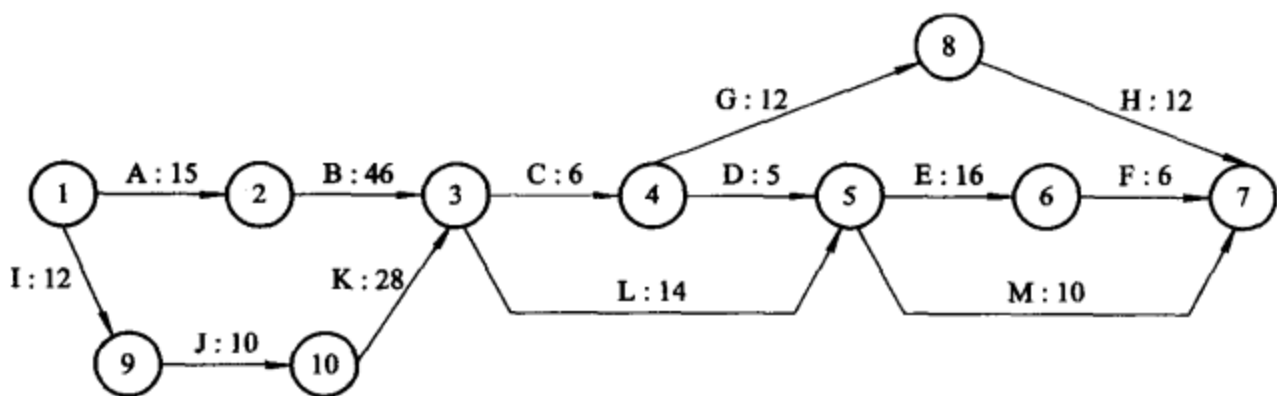


图 1-11 保守情况下任务进度图

图 1-11 中同样存在 ABCDEF、ABCDM、ABCGH、ABLEF、ABLM、IJKDEF、IJKDM、IJKCGH、IJKLEF、IJKLM 等 10 条路径。假设图 1-11 中箭头上的时间量是确定性的估计值, 那么路径 IJKLEF 需花费 $12 + 10 + 28 + 14 + 16 + 6 = 86$ 天, 而路径 ABLEF 需花费 $15 + 46 + 14 + 16 + 6 = 97$ 天。根据关键路径的定义可知, 路径 ABLEF 是保守情况下该软件开发任务进度的关键路径。

由以上分析过程可知, 在保守情况下, 需要 97 天才能完成此工控领域的嵌入式软件开发任务。

【问题 3】(2 分)

本题的分析思路请参考以上【问题 2】的要点解析, 在最理想的情况下图 1-9 的关键路径是 IJKLEF; 在保守情况下图 1-9 的关键路径是 ABLEF。

【问题 4】(2 分)

为了加速项目开发进度, 需抓紧最理想和保守的估计中的共同阶段任务的时间安排。最理想的情况下图 1-9 的关键路径是 IJKLEF, 而其保守情况下的关键路径是 ABLEF, 在这两种情形的共同阶段任务是 L、E、F(关键路径的交集), 即用户各类操作和使用方面的培训、优化调整源程序循环、用户确认生效这三个阶段。

【问题 5】(3 分)

这是一道要求读者掌握嵌入式系统开发中项目管理基础知识的识记题。本题的解答思路如下:

① 理解题干给出的“项目管理”的概念, 该定义中需重点理解两个短语, 即定义中“实现项目全过程的动态管理”是指在项目的生命周期内, 不断进行资源的配置和协调, 不断做出科学决策, 从而使项目执行的全过程处于最佳期的运行状态, 产生最佳的效果。另外, 定义中“项目目标的综合协调与优化”是指项目管理应综合协调好时间、费用及功能等约束性目标, 在相对较短的时期内成功地达到一个特定的成果性目标。

② 一般来讲, 项目管理是按任务(垂直结构)而不是按职能(平行结构)组织起来的。项目管理的主要任务一般包括项目计划、项目组织、质量管理、费用控制、进度控制五项。日常的项目管理活动通常是围绕这五项基本任务展开的。项目管理贯穿于一个项目的整个生

命周期,它是一种运用既规律又经济的方法对项目进行高效率的质量考核,并注重将当前的执行情况与前期进行比较。

③ 对于嵌入式软件项目的开发,项目进度管理、项目范围管理、项目成本管理、项目配置管理、项目质量管理、项目风险管理、开发人员管理等七个方面的管理工作极为重要。项目开发的每个阶段、每个过程都要重视这几个方面的管理。

三、参考答案

表 1-16 给出了本案例问题 1 至问题 5 的参考答案,供读者练习时参考,以便查缺补漏。读者也可依照所给出的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

表 1-16 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (3 分)	优点:关注满足客户需求,减少由于软件需求不明确带来的开发风险(1 分) 缺点:可能导致系统设计差、效率低、难以维护(2 分)	
【问题 2】 (5 分)	<p>最理想情况 82 天完成(3 分)</p> $T_B = \sum_{i=1}^4 (T_{B1} + T_{B2} + T_{B3})$ $= (5 + 10 + 3) + (3 + 4 + 2) + (2 + 2 + 1) + (1 + 1 + 0) = 34 \text{ 天}$ $T_E = \sum_{i=1}^2 (T_{E1} + T_{E2}) = (8 + 2) + (2 + 0) = 12 \text{ 天}$ <p>路径 JKLEF 需花费 $12 + 10 + 28 + 14 + 12 + 6 = 82 \text{ 天}$ 保守估计 97 天完成(2 分)</p> $T_B = \sum_{i=1}^8 (T_{B1} + T_{B2} + T_{B3})$ $= (5 + 10 + 3) + (3 + 4 + 2) + (2 + 2 + 1) + (1 + 1 + 0) + 4 \times (1 + 1 + 1)$ $= 34 + 12 = 46 \text{ 天}$ $T_E = \sum_{i=1}^3 (T_{E1} + T_{E2}) = (8 + 2) + (2 + 2) + (2 + 0) = 16 \text{ 天}$ <p>路径 ABLEF 需花费 $15 + 46 + 14 + 16 + 6 = 97 \text{ 天}$</p>	
【问题 3】 (2 分)	在最理想的情况下关键路径是 JKLEF(1 分) 在保守情况下关键路径是 ABLEF(1 分)	
【问题 4】 (2 分)	要着重抓的共同环节是 L、E、F,即用户各类操作和使用方面的培训、优化调整源程序循环、用户确认生效这三个阶段(2 分)	
【问题 5】 (3 分)	项目范围管理、项目成本管理、项目配置管理、项目质量管理、项目风险管理、开发人员管理等(每小点 0.5 分)	

1.2.5 案例 5 UML 建模技术

一、案例描述

在目前的软件开发方法中,面向对象的方法占据着主导地位。而统一建模语言 UML

(Unified Modeling Language)是面向对象技术领域内主流的标准建模语言，它融入了软件工程领域的新思想、新方法和新技术，不仅可以支持面向对象的分析与设计，更重要的是能够有力地支持从需求分析开始的软件开发全过程，是一种定义良好、易于表示、功能强大且普遍实用的建模语言。

阅读下列关于某嵌入式门禁控制系统的技术说明，根据要求回答问题 1 至问题 4。
(15 分)

【说明】

门禁系统是楼宇安防系统的重要组成部分，也是大厦智能化管理的体现。性能可靠、操作简便、技术先进、扩展性强、维护方便是嵌入式门禁系统产品的设计思想。某感应门禁系统组成结构如图 1-12 所示。

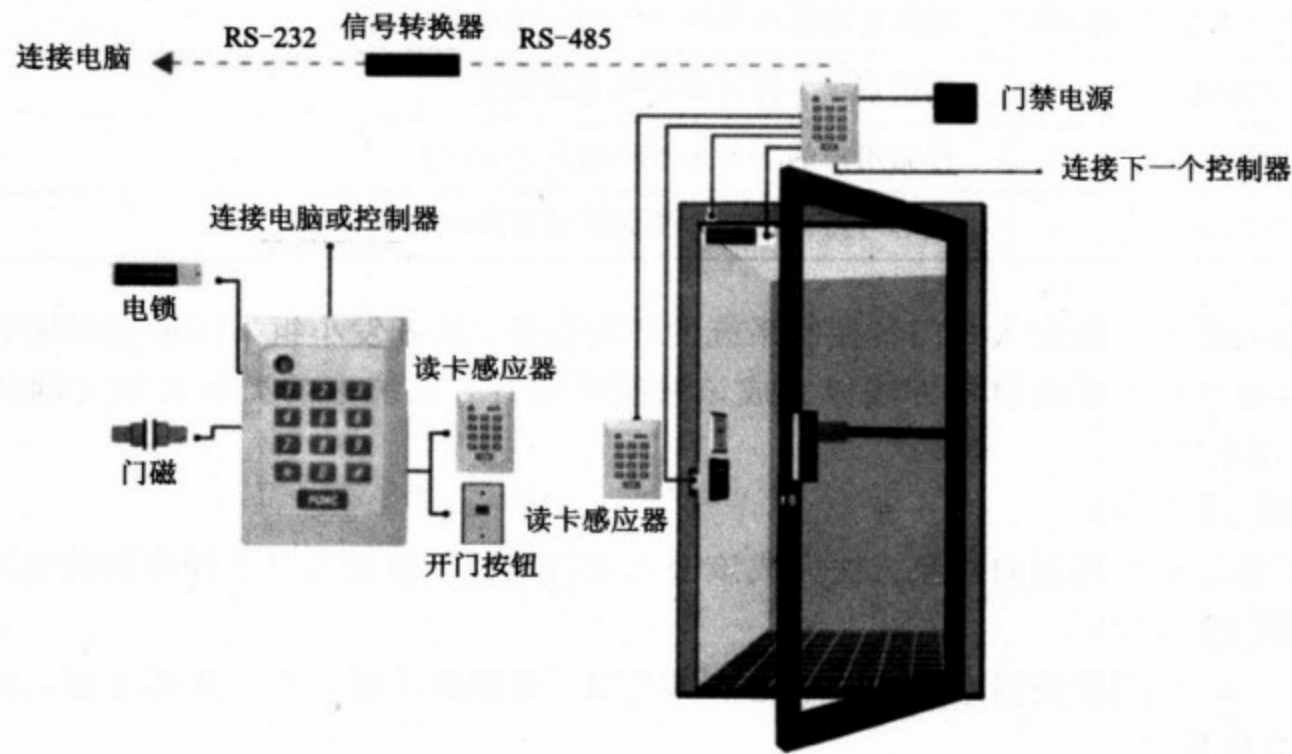


图 1-12 某感应门禁系统组成结构示意图

通常要求准备进入的人员在入口处刷卡，锁控器(LockController)接收读卡器(Finger-Reader)的读卡信息并进一步识别，如果为有效卡，则触发控制电控锁(Lock)的继电器，让持卡人通过。锁控器还会将这些读卡及进出事件记录存储起来，并将相关事件记录传送给上位机。每个锁控器管理 1~4 个门，每 5~8 个锁控器接入 1 条 RS-485 总线，并转换成 RS-232 方式与管理主机的串行口。对锁控器进行相关设置后，允许它脱机独立对门禁点进行控制，管理主机或通信线路故障都不会影响它的正常运行。在普通场合通常会设置 1 个出门按钮，允许已进入的人员按动此按钮以打开电控锁；而对于一些重要场合通常会在门的内外安装 2 个读卡器，进出门时都需按预设方式进行刷卡。

门禁系统中的每个电控锁都有一个惟一的编号。锁的状态有“已锁住”和“未锁住”两种。在管理主机上可以设置每把锁的安全级别以及用户的开锁权限。只有当用户的开锁权限大于或等于锁的安全级别并且锁处于“已锁住”状态时，才能将锁打开。用户的感应卡信息、开锁权限以及锁的安全级别都保存在管理主机的数据库中。

表 1-17 给出了图 1-12 结构图中主要组成部件的作用。

表 1-17 感应门禁系统主要组成部件及其作用

部件名称	主要作用(功能)
读卡感应器 (FingerReader)	通过射频感应原理,识别感应卡内置加密卡号
锁控器(LockController)	存储感应卡权限和刷卡记录,向管理软件上传读卡器送来的信号,并负责和上位机通信和其他数据存储器协调
电控锁(Lock)	电动执行机构
主机管理软件 (Win-Pak Software)	通过电脑对所有单元进行中央管理和监控,进行相应的时钟、授权、统计管理工作
RS-485/232 信号转换器	对所有数据存储器进行联网和远距离通信
门禁电源	提供系统工作时所需的电源能量
感应卡	存储用户的不可复制和解密的 ID 号
开门按钮	出门时可以设置为按此按钮出门(可选部件)

A 公司承担了该嵌入式门禁控制系统的开发任务,其开发小组采用根据问题领域的模型建立系统结构的面向对象方法以完成该系统的设计,系统中的类以及类之间的关系用 UML 类图来表示。

【问题 1】(4 分)

请简要说明采用面向对象方法完成该嵌入式门禁控制系统的开发任务的优越之处。

【问题 2】(3 分)

图 1-13 是门禁控制系统一个不完整的类图,根据题干说明中给出的术语,请给出类 Lock 的主要属性。

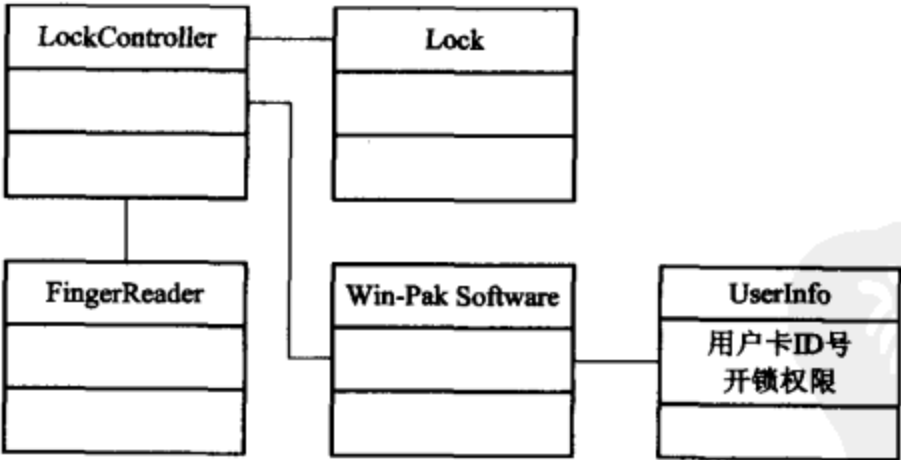


图 1-13 门禁控制系统类图

【问题 3】(5 分)

序列图是场景的图形化表示,描述了以时间顺序为组织的对象之间的交互活动,即显示了一组对象和由这组对象发送和接收的消息。图 1-14 是用户成功开锁的序列图。依据上述说明中给出的词语,将图 1-14 中的(1)~(5)空缺处补充完整。

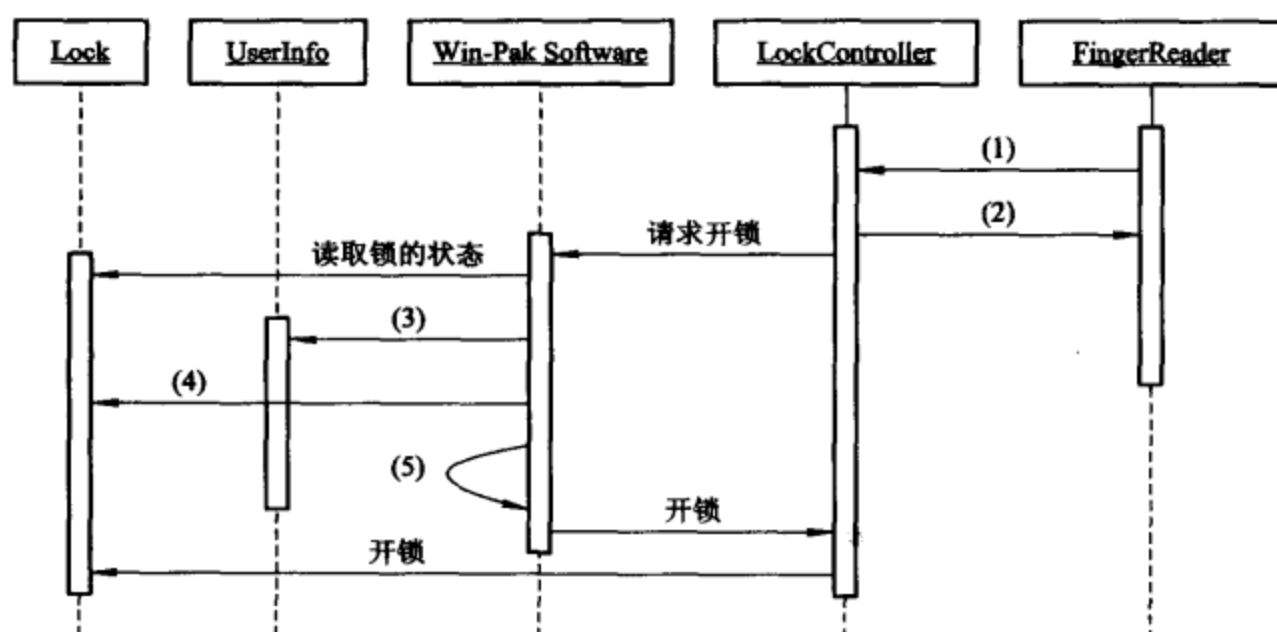


图 1-14 用户成功开锁的序列图

【问题 4】(3 分)

随着嵌入式应用的不断增长，嵌入式系统需求的复杂性、不确定性不断提高，系统规模也逐步扩大，这给嵌入式应用软件的开发带来了新的挑战。同时，嵌入式软件的开发又必须面对由于芯片性能的增长、嵌入式操作系统平台等技术方面不断变化所带来的各种压力。一种基于 UML 嵌入式软件开发环境(如图 1-15 所示)的发展，使一直“深埋”于系统的嵌入式应用软件变得开放而易于开发，大大缩短了产品的开发周期，解决了嵌入式应用软件的移植问题，使软件的开发工作主要集中在高层的建模和模型的测试及验证上，从而使软件开发工作的焦点从编码转到了设计上来。

请将图 1-15 中(6)~(8)空缺处填写完整，从而得到一张完整的基于 UML 嵌入式软件开发环境的结构图。

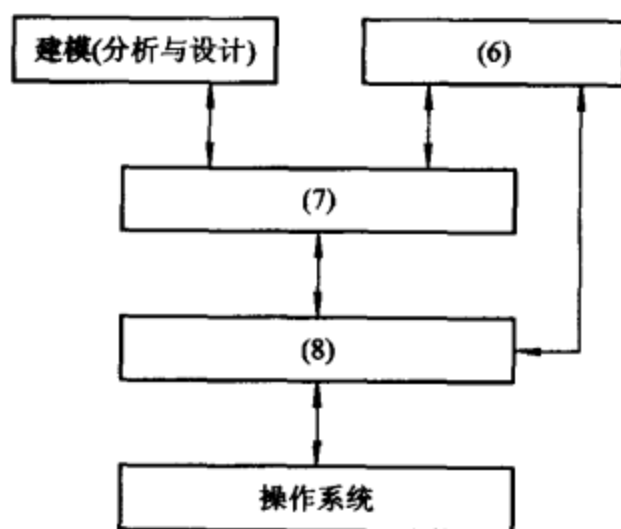


图 1-15 基于 UML 嵌入式软件开发环境的结构图

二、要点解析**【问题 1】(4 分)**

这是一道要求读者说明面向对象方法在嵌入式系统开发中优越性的简答试题。本题的解答思路是，根据 James Rumbaugh 的定义可以得出建模是为了捕捉、描述系统的核心。

由于对象是数据及其操作的封装体,在进行嵌入式系统开发时,应用面向对象的方法相对于传统的开发方法的优越性主要体现在以下几个方面:

① 符合人类的思维习惯。面向对象的方法就是按照人们习惯的思维方式建立问题模型,它为开发者提供了随着对某个应用系统的认识逐步深入和具体化的过程。由于可以先设计出由抽象类构成的系统框架,随着认识的深入和具体化再逐步派生出更具体的派生类,因此这种开发过程符合人们认识客观世界解决复杂问题时逐步深化的渐进过程。

② 稳定性好。由于面向对象的嵌入式系统的体系结构是根据问题领域的模型建立起来的,而不是基于对系统应完成的功能的分解,因此当对系统的功能发生变更时并不会引起整个系统体系结构的整体变化,仅需要做一些局部的修改。

③ 可重用性好。由于对象具有固有的封装性和继承性,使得对象内部的实现与外界隔离,具有较强的独立性。因此采用面向对象方法完成系统设计可以提高所开发模块的可重用性。

④ 可维护性好。由于面向对象方法的继承性和可重用性使得嵌入式系统开发模块化,易于理解其功能结构。正因为它的稳定性和模块的独立性使得软件容易修改、测试和调试。

【问题 2】(3 分)

这是一道要求读者掌握如何从问题域中抽象出类的属性的综合分析题。本题的解答思路如下:

① 为了便于读者对门禁控制系统的组成部件的物理结构有所了解,图 1-16 先给出了市场中常见的一组门禁系统部件。对于本问题的解答首先要仔细理解题目中的信息,从这些信息中获知该感应门禁系统主要部件有锁控器(LockController)、读卡感应器(Finger-Reader)、电控锁(Lock)和主机管理软件(Win-Pak Software)。将关键信息转化成如图 1-17 所示的系统体系结构。



图 1-16 一组常见的门禁系统部件

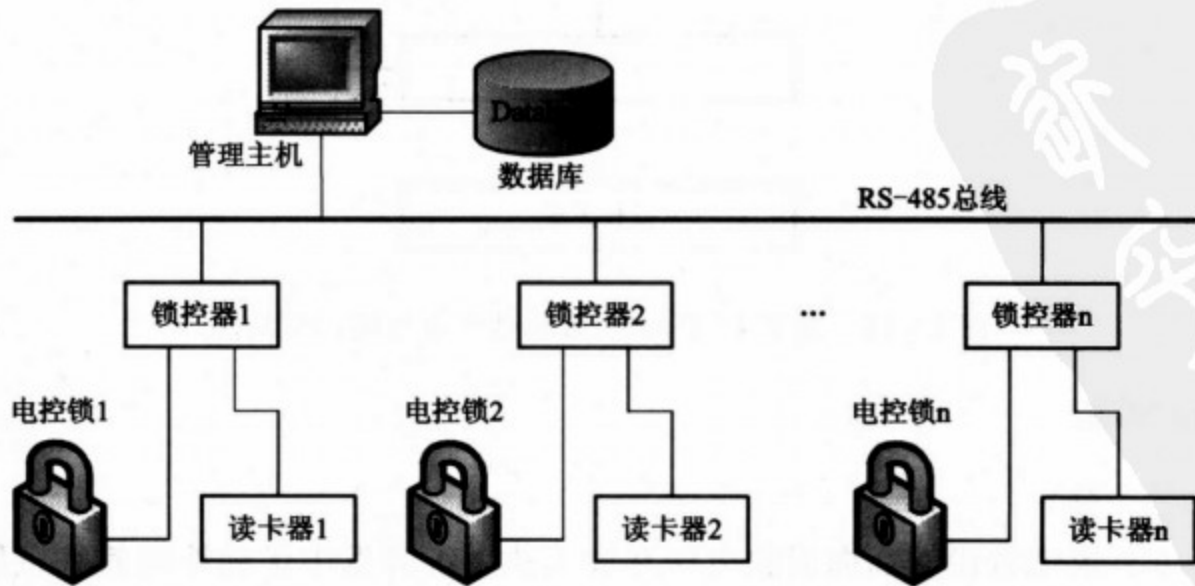


图 1-17 感应门禁系统体系结构图

② 在面向对象系统建模过程中，通常用概念模型来详细描述系统的问题域。类图用于描述系统的结构化设计，即用来表示概念模型，它最基本的元素是类或接口，表达了类、接口以及它们之间的静态结构和关系。在类图上通常包含表 1-18 中所给出的六种关系。

表 1-18 类图的六种关系

序号	关 系	说 明	箭 头 方 向	关键字
1	泛化 (Generalization)	表示类与类、接口与接口之间的继承关系	由子类指向父类	extends
2	实例 (Realization)	用于指定两个实体之间的一个合同	由实现接口的类指向被实现的接口	implements
3	关联 (Association)	表示类与类之间的连接，即一个类保存对另一个类实例的引用，并在需要时调用这个实例的方法	单向箭头指向遍历或查询的方向；而双向箭头是可选的	
4	聚集 (Aggregation)	是关联的一种形式，代表两个类之间的整体/局部关系		
5	组装 (Composition)	是聚集的一种特殊形式，暗示“局部”在“整体”内部的生存期职责		
6	依赖 (Dependency)	也是类与类之间的连接，并且依赖总是单向的，是类间最弱的一种关系		

注意在表 1-18 中，聚集暗示着整体在概念上处于比局部更高的一个级别，在实例图中不存在回路，即只能是一种单向关系，而关联暗示两个类在概念上位于相同的级别。在 Java 中，关联使用实例变量来实现，聚集也是使用实例变量来实现的。关联和聚集的区别纯粹是概念上的，在 Java 语法上没有刻意的区分。

另外，聚集和组装是 UML 中两种非常重要的关系，它们都表示实例之间的整体/部分关系。组装是聚集的一种形式。聚集是概念性的，只是区分整体与部分。组装具有很强的归属关系，而且整体与部分的对象生存周期是一致的。对于这一知识点也可理解成如果没有成分对象，组装对象就不存在；在任何时候，每个给定的成分对象只能是组装对象的组成部分。

③ 从图 1-13 中可以看出，类图是面向对象建模技术中最常见的图形，它显示了一组类、接口、协作以及它们之间的相互关系。当对系统的静态设计视图建模时，通常以对系统的词汇建模、对简单协作建模、对逻辑数据库模式建模这三种方式之一使用类图。

其中，对系统的词汇建模需划分哪些抽象是考虑中的系统的一部分，哪些抽象是处于系统的边界之外，即用类图详细描述这些抽象和它们的职责。而在简单协作建模中，协作是一些共同工作的类、接口和其他元素的群体，该群体提供的一些合作行为强于所有这些元素的行为之和。在很多领域中要在关系数据库或面向对象数据库中存储永久信息，可以用类图对这些数据库的模式进行建模。

④ 图 1-13 类图中已经完成了面向对象分析中的认定类，读者所要完成的工作是定义

类的内部信息。类 Lock 是本系统中的一个关键类，题目与它的属性相关的描述有“门禁系统中的每个电控锁都有一个惟一的编号”、“锁的状态有两种”、“在管理主机上可以设置每把锁的安全级别”、“锁处于‘已锁住’状态时，才能将锁打开”等。从这些信息中提炼出说明类 Lock 关键属性及特性的答案，即“锁的编号”、“锁的状态”和“锁的安全级别”等。

【问题 3】(3 分)

这是一道要求读者掌握 UML 建模中序列图知识点的分析题。本题的解答思路如下：

① 系统的动态行为常采用 UML 序列图表示，它展现了一组对象和由这组对象收发的消息，用于按时间顺序对控制流建模。注意它强调的是时间和顺序。

② 图 1-14 所表达的序列图有两个特征：一是它有对象生命线；二是它有控制焦点。对象生命线是一条垂直的虚线，表示一个对象的生命跨度。控制焦点是一个瘦高的矩形，表示一个对象执行一个动作所经历的时间段。矩形的顶部表示动作的开始，底部表示动作的结束。在序列图顶部水平方向画出的是参与交换的对象。

③ 看懂序列图之后，接着根据问题要点，查找题目中与用户开锁相关的信息。对这些关键信息进行归纳整理的结果如下：用户开锁时，只需将感应卡靠近读卡器。读卡器获取信息后将发送一个中断事件给锁控器，锁控器从读卡器读取用户卡的 ID 号并将该 ID 号信息发送到管理主机，管理主机根据数据库中存储的信息来判断用户是否具有开锁权限，若有且锁当前处于“已锁住”状态，则将锁打开；否则系统报警。

④ 通过以上分析，可以得出用户开锁的激发事件是：用户感应卡靠近读卡器。

⑤ 由于序列图是按照时间顺序组织对象之间交互活动，因此需要将这些活动按照时间顺序进行排序，并记录下参与每个活动的对象。根据以上分析结果结合图 1-14 已给出的信息，可以得出用户开锁涉及的活动事件如表 1-19 所示。

表 1-19 用户开锁活动过程

顺序	活动事件	活动对象
1	发送“中断事件”	读卡器→锁控器
2	读取用户卡的 ID 号	锁控器→读卡器
3	请求开锁	锁控器→管理主机
4	读取锁的当前状态	管理主机→锁
5	读取用户的开锁权限	管理主机→用户
6	读取锁的安全级别	管理主机→锁
7	判断用户是否能够开锁	管理主机→管理主机
8	通知能够开锁	管理主机→锁控器
9	将锁打开	锁控器→锁

【问题 4】(3 分)

这是一道要求读者熟悉典型的 UML 嵌入式软件开发环境的试题。本题所涉及的知识如下：

① 由 I-Logix 公司开发的 Rhapsody 是一个基于 UML 的面向嵌入式实时应用开发的集成、可视化环境。软件开发者可以在这个环境里进行分析、设计、实现及验证。

② Rhapsody 支持基于模型的调试；提供专门为实时嵌入式应用设计的可执行的框架，可以产生基于 VxWorks、POS 和 OSE 等多种操作系统的 C 语言、C++ 语言和 Java 语言的源程序。

③ 图 1-18 给出了一张完整的基于 UML 嵌入式软件开发环境的结构图，虚线框部分表示基于 UML 的软件开发环境。

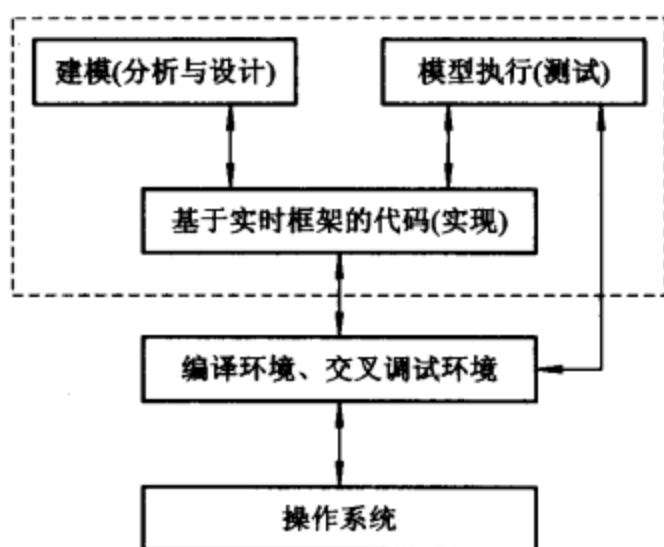


图 1-18 完整的基于 UML 嵌入式软件开发环境的结构图

④ 在图 1-18 中“建模(分析和设计)”是用 UML 来描述，对系统建模。

⑤ “基于实时框架的代码(实现)”这一过程利用代码自动生成技术来实现。

⑥ “模型执行(测试)”这一过程将依赖于生成的代码，通过在代码中拆装一些用于支持模型调试的调试信息来实现。

⑦ 目标系统的操作系统开发环境来完成代码的编译和链接任务。

⑧ 代码的运行与源程序级的调试仍然采用一般的嵌入式软件调试环境。

另外补充一点，在目前的软件开发方法中，面向对象的方法占据着主导地位，这种主导地位也决定着软件开发过程模型化技术的发展。根据对目前软件业的研究和估计，统一建模语言 UML 可以说代表着 21 世纪初期软件建模的发展方向。UML 共定义了结构类、行为类和模型管理类三个大类，共 12 种模型图，详见表 1-20 所归纳的内容。

从应用的角度看，当采用面向对象技术设计系统时，首先是描述需求；其次是根据需求建立系统的静态模型，以构造系统的结构；第三步是描述系统的行为。第一步与第二步中所建立的模型都是静态的，包括用例图、类图(包含包)、对象图、组件图和配置图这五种图形，是标准建模语言 UML 的静态建模机制。第三步中所建立的模型或者可以执行，或者表示执行时的时序状态或交互关系，它包括状态图、活动图、顺序图和协作图这四种图形，是标准建模语言 UML 的动态建模机制。因此，标准建模语言 UML 的主要内容也可以归纳为静态建模机制和动态建模机制两大类。

表 1-20 UML 模型图

类型	功能	子类型	功 能
结构类 模型图	描述 系统应 用的静 态结构	类图	描述系统中类的静态结构，展示了一组类、接口和协作及它们间的关系，描述的是一种静态关系，在系统的整个生命周期都是有效的。系统可有多个类图，在高层给出类的主要职责，在低层给出类的属性和操作
		对象图	展示了一组对象及它们间的关系，用对象图说明类图中所反映事物实例的数据结构和静态快照，是类图的显示类的多个对象实例，而不是实际的类，只能在系统某一段时间段存在
		组件图	描述代码组件的物理结构及各组件之间的依赖关系，用于对源代码、可执行的发布、物理数据库和可调整的系统建模
		配置图	展现了运行时处理节点及其构件的部署，它描述系统硬件的物理拓扑结构(包括网络布局和构件在网络上的位置)以及在此结构上执行的软件(即运行时软构件在节点中的分布情况)。它说明系统结构的静态部署视图，即说明分布、交付和安装的物理系统
行为类 模型图	描述 系统动 态行为 的各个 方面	用例图	展现了一组用例、用户以及它们间的关系，即从用户角度描述系统功能，并指出各功能的操作者。用于收集用户实际需求所采用的一些方法中
		序列图	展现了一组对象和由这组对象收发的消息，用于按时间顺序对控制流建模，用它说明系统的动态视图。强调的是时间和顺序
		活动图	是一种特殊的状态图，描述需要做的活动、执行这些活动的顺序(多为并行的)以及工作流(完成工作所需要的步骤)。强调对象间的控制流程
		协作图	展现了一组对象及其相互间的连接，还有这组对象收发的消息。它强调收发消息对象的结构组织，按组织结构对控制流建模。它强调上下层次关系
		状态图	展示了一个特定对象的所有可能状态以及由于各种事件的发生而引起的状态间的转移，常用它说明系统的动态视图。它对于接口、类或协作的行为建模尤为重要。一个状态图描述了一个状态机，可用它描述使用用例的生命周期
模型管 理类模 型 图	组织 和管理 各种应 用模型	软件包	
		子系统	
		模型	

三、参考答案

表 1-21 给出了本案例问题 1 至问题 4 的参考答案，供读者练习时参考，以便查缺补漏。读者也可依照所给出的评分标准得出自己的测试分数，从而大致评估自己对这一方面知识点的掌握程度。

表 1-21 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (4 分)	① 符合人类的思维习惯 ② 稳定性好 ③ 可重用性好 ④ 可维护性好(每小点 1 分)	
【问题 2】 (3 分)	① 锁的编号 ② 安全级别 ③ 锁的当前状态(每小点 1 分)	
【问题 3】 (5 分)	(1) 中断事件 (2) 读取用户卡的 ID 号 (3) 读取用户开锁权限 (4) 读取锁的安全级别 (5) 判断用户是否有权限开锁，或用户是否可以开锁(每空 1 分)	
【问题 4】 (3 分)	(6) 模型执行(测试) (7) 基于实时框架的代码(实现) (8) 编译环境、交叉调试环境(每空 1 分)	

1.3 真题链接

1.3.1 2003 年系统分析员级下午试卷 I 试题 1

一、试题描述

阅读以下关于嵌入式系统设计中的软硬件功能划分的说明，回答问题 1 至问题 3。
(25 分)

【说明】

图 1-19 为一个 SOC(System On Chip, 片上系统)设计中的控制流程图。该设计的约束条件是：在时钟周期为 10 ns 的情况下，系统的延迟时间小于等于 4000 ns (系统由一个处理器和若干个运算部件构成)。

表 1-22 给出了可供使用的硬件库，其中可选的器件有乘法器(Mul)、加法器(Add)以及比较运算器(Comp)。每个被选器件的名称、延时时间以及价格等如该表所示。

可选的处理器分别有 Pentium、PowerPC 和 6800 三类，每类处理器的相关指标如表 1-23 所示。

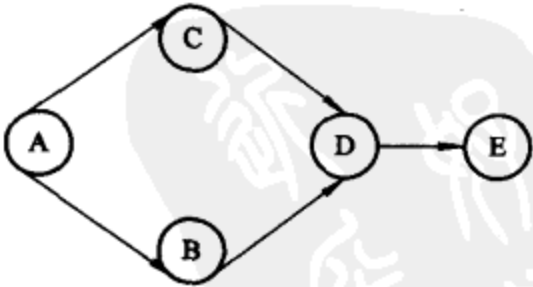


图 1-19 某 SOC 的控制流程图

经过测试, 流程图中各个节点的软件执行时间如表 1-24 表示, 构成各个节点的器件(各节点内器件可以串联或并联工作)如表 1-25 所示。

表 1-22 硬 件 库

类 别	名 称	延时时间(ns)	门 数	价 格(元)
*	Mul1	30	100	90
*	Mul2	50	70	70
*	Mul3	70	60	50
+	Add1	30	45	35
+	Add2	42	30	30
>	Comp1	18	12	15
=	Comp2	14	8	10

表 1-23 处 理 器

类 别	时 钟 周 期(个)	价 格(元)
Pentium	10	900
PowerPC	10	750
6800	50	600

表 1-24 节点的软件执行时间

节 点	处 理 器	执 行 时 间(ns)
A	Pentium	3100
	PowerPC	3800
	6800	6000
B	Pentium	1400
	PowerPC	2200
	6800	2800
C	Pentium	6800
	PowerPC	12 000
	6800	18 000
D	Pentium	900
	PowerPC	1000
	6800	1200
E	Pentium	12 000
	PowerPC	14 800
	6800	21 500

表 1-25 器 件 表

节 点	构 成 的 硬 件
A	2 个 Mul1, 2 个 Mul3, 2 个 Add2, 2 个 Comp1
B	2 个 Mul2, 3 个 Add1, 1 个 Add2
C	2 个 Mul1, 1 个 Mul3, 2 个 Add2, 2 个 Comp2
D	2 个 Mul2, 2 个 Add1, 1 个 Add2, 2 个 Comp1
E	1 个 Mul1, 1 个 Mul2, 2 个 Add1, 2 个 Comp2

【问题 1】(5 分)

为了满足系统设计的要求,并使最终产品成本较低,图 1-19 中的哪几个节点必须用硬件实现?

【问题 2】(10 分)

请给出图 1-19 的最佳软件实现方案,并说明原因。

【问题 3】(10 分)

如果系统延迟时间改为不超过 3500 ns,则软硬件的实现方案如何?请在 100 字以内给予说明。

二、参考答案**【问题 1】(5 分)**

为了满足系统设计的要求,并使最终产品成本较低,图 1-19 中节点 A、C、E 同时用硬件实现(3 分),而节点 B、D 可同时用软件实现(2 分)。

【问题 2】(10 分)

图 1-19 的最佳软件实现方案,节点 A、C、E 同时用硬件实现(3 分),而节点 B、D 可同时用软件实现(2 分)且需采用 PowerPC 处理器来使系统成本最低(2430 元,2 分),此时整个系统流程的延迟时间为 3688 ns(3 分)。

【问题 3】(10 分)

如果系统延迟时间改为不超过 3500 ns,则节点 A、C、E 都必须用硬件实现(3 分),且 B、D 不能同时采用 PowerPC 处理器(2 分),节点 B 采用 Pentium 处理器(1 分),节点 D 采用 PowerPC 处理器(1 分),此时整个系统流程的延迟时间为 $320+1400+1000+168=2888\text{ ns}<3500\text{ ns}$ (2 分),系统成本为 $370+900+310+750+250=2580$ 元(1 分)。

1.3.2 2005 年下半年系统分析员级下午试卷 I 试题 3**一、试题描述**

阅读以下关于某嵌入式实时系统的软件需求的说明,回答问题 1 至问题 3。(25 分)

【说明】

某公司承担了一项嵌入式实时控制系统的软件开发任务,其内容是按系统要求的固定的时间序列采集、处理和输出数据,以实现对多个设备的综合控制。

系统由硬件和软件组成,硬件由处理机(采用 PowerPC603e,主频 133 MHz)、存储器、定时器、中断控制器、双口存储器(空间大小 1024 KB)、四路 422 半双工串口接口(其中两路的频率为 115 200 Hz,另两路频率为 38 400 Hz)、两路 A/D、D/A 数模转换器和十路离散量接口组成。系统的软件需求如下(注:B 为字节;s 为秒;ms 为毫秒):

(1) 5 ms 任务:系统要求以 5 ms 为周期从双口存储器中采集 1024 KB 的输入数据,处理任务的时间约为 1 ms。

(2) 20 ms 任务:系统要求以 20 ms 为周期从两路 422 接口(115 200 Hz)中采集一定格式的大小为 64 B 的数据,在完成处理(处理时间约为 4 ms)任务后,分别输出大小为 16 B 的控制命令。

(3) 60 ms 任务:系统要求以 60 ms 为周期从两路 422 接口(384 00 Hz)中采集一定数据格式的大小为 6B 数据,从两路 A/D 接口采集 28 位数据,在完成处理(处理时间约为 2 ms)任务后,分别输出大小为 2B 的控制命令和一个 28 位 D/A 数据,输出八路离散量控制数据。

(4) 1 s 任务:作为系统的安全监控保障,系统要求在每 1 s 内对系统软硬件状态进行测试并完成系统工作的状态记录工作。该任务共需处理时间是 5 ms。

本公司课题组根据用户的硬件环境及软件的需求,就软件的实施方案展开了激烈的讨论。讨论的首要问题就是本系统中软件的运行平台是采用嵌入式实时操作系统还是在裸机上直接开发的方式。李工程师(简称李工)提出,为保证系统的实时性,应该采用裸机上直接开发的方式,关键部分采用汇编语言编写;而王工程师(简称王工)提出,由于系统将由多种周期的处理任务组成,为了保证系统的可靠性,应该采用商用嵌入式实时操作系统作为本项目的开发基础。经过充分论证,最后李工接受了王工的建议,并申报课题组长,课题组长同意采用了王工的方案,并指出必须尽快对需求的时间性能做出评估。

【问题 1】(10 分)

请在 450 字以内简要说明王工提出的采用嵌入式实时操作系统的理由及优点,并说明选择操作系统产品时需要重点考虑哪些功能与性能。

【问题 2】(10 分)

李工和王工对需求进行了认真分析,给出了时间性能评估报告,判定在上述硬件平台上可以满足系统要求。请在 300 字以内简要说明嵌入式实时操作系统中时间性能评估中主要考虑哪几项因素,并针对本课题的具体数据,叙述系统的工作时序关系。

【问题 3】(5 分)

根据系统需求,请设计出系统的处理流程(按时序),请在 350 字以内简要说明所有任务的优先级分配策略,422 接口、双口存储器、离散量接口和 A/D(D/A)的数据输入/输出方法(中断或查询驱动)及理由。

二、参考答案

【问题 1】(10 分)

采用嵌入式实时操作系统的主要理由及优点:

操作系统提供的任务调度功能,可以有效地管理本系统多周期任务的调度,并且操作系统提供的事件、信号和任务间的通信机制,可以有效地解决本系统中资源共享中的互斥问题。

采用商品化的软件,可提高所开发软件的可靠性,还可简化软件开发成本,提高开发效率。

选择嵌入式实时操作系统,主要考虑操作系统的以下功能和性能:

① 任务的上下文切换时间越短越好,中断响应时间越快越好,内核代码占用空间比较小。

② 操作系统提供的服务接口应较丰富,有利于灵活使用(接口的开放性)。

③ 操作系统对可靠性有较强的保障支持能力。

④ 使用简单,支撑环境配套好,可配置和可剪裁能力强。

【问题 2】(10 分)

嵌入式实时系统中时间性能评估主要考虑以下几个因素:

① 系统(操作系统)时间开销,一般不大于整个控制系统运行时间总开销的 20%。

② 保证所有任务在规定的时间内完成(或进行可调度性评估)。

③ 任务的上下文切换时间和中断响应时间。

针对本题的具体数据,系统的工作时序关系应该是:

系统的最小时间节拍定义为 5 ms,系统最大工作周期(或称主时间框架)为 1 s。在每个 5 ms 起点,5 ms 任务首先运行,占 1 ms 时间,完成后转入处理被终止的任务。如果这个时刻时间到达 20 ms 的起点,则启动 20 ms 任务运行。以此类推,确保在 1 s 周期内所有任务按规定的时间序列执行。

【问题 3】(5 分)

系统的处理流程如图 1-20 所示。

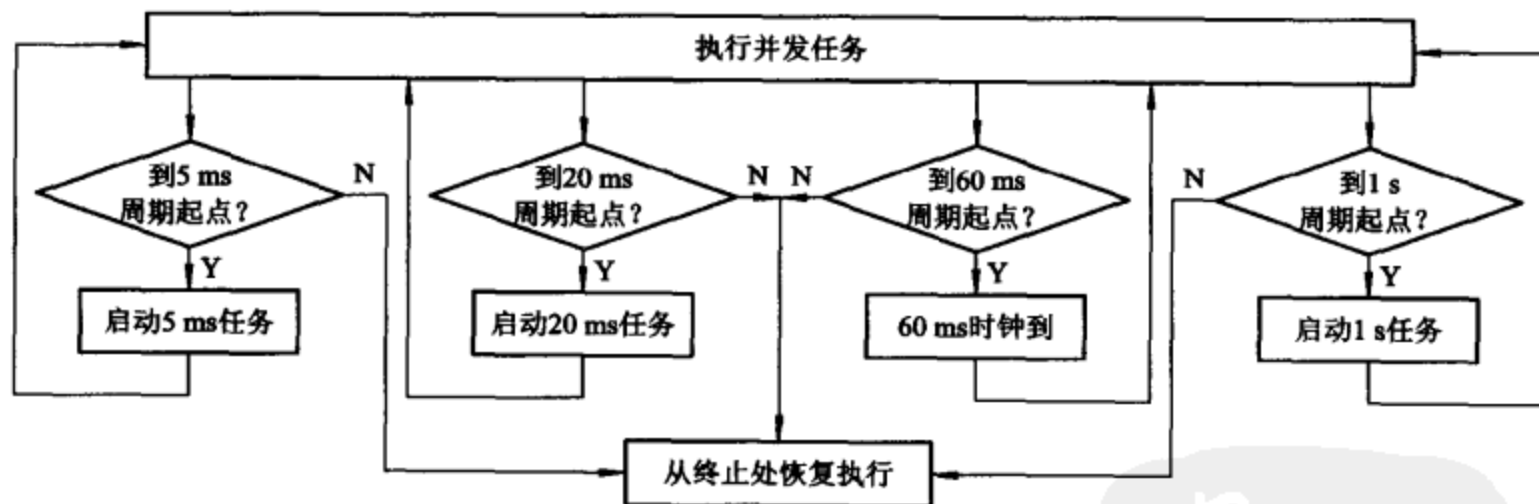


图 1-20 系统的处理流程图

优先级分配策略:小周期优先策略。

任务优先级从低到高排序是:

1 s 任务 60 ms 任务 20 ms 任务 5 ms 任务

最低 ----- > 最高

数据输入/输出处理方式:422 接口适合采用中断方式,主要因为处理机速度远远比数据传输速度快,在采集时不易消耗太多的时间,而查询方式由于存在等待数据时间,消耗处理机时间较大。双口存储器和离散量接口由于是直接访问存储器,故采用查询较合适。A/D 和 D/A 存在数据的转换时间,应在等待时交出处理机时间(定时查询)。

1.3.3 2006 年下半年系统分析员级下午试卷 I 试题 2

一、试题描述

阅读以下关于嵌入式实时操作系统软件的选型分析方面的说明，回答问题 1 至问题 3。
(25 分)

【说明】

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的(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、信号处理机以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固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 1 s，最小周期为 5 ms。数据处理机上任务分布如表 1-26 所示。

表 1-26 数据处理机上任务分布表

工作周期(个)	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(例如 Vx-Works 等)；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】(10 分)

为什么专家组一致认为王工的实施方案切实可行？请在 200 字以内简要说明。

【问题 2】(7 分)

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？请在 200 字以内简要论证。

【问题 3】(8 分)

根据本项目中任务的时间性能参数，请在 200 字以内(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)。

二、参考答案**【问题 1】(10 分)**

本系统各周期任务的工作时序需在规定的时间内完成，偶尔出现违反某些应用的时间需求将对系统运行造成严重影响。硬实时处理能力的嵌入式实时操作系统提供的任务调度功能，可以有效地管理本系统多周期任务的调度，并且操作系统提供的事件、信号和任务间的通信机制，可以有效地解决本系统资源共享中出现的互斥问题。

【问题 2】(7 分)

选择实时嵌入式操作系统及开发环境时，主要考虑以下几方面的因素：

- ① 操作系统是否支持目标硬件平台，操作系统的可移植性和可扩充性；
- ② 操作系统任务的上下文切换时间越短越好，中断响应时间越快越好，内核代码占用空间比较小；
- ③ 操作系统提供的服务接口应较丰富，有利于灵活使用(接口的开放性)；
- ④ 使用简单，开发工具的支持程序配套好，可配置和可剪裁能力强；
- ⑤ 应用对操作系统性能的要求，中文内核支持，标准兼容性，技术支持，源代码还是目标代码；
- ⑥ 操作系统对可靠性有较强的保障支持能力。

【问题 3】(8 分)

针对本题的具体数据，系统任务的被调度关系应该是：

系统的最小时间节拍定义为 5 ms，系统最大工作周期(或称主时间框架)为 1 s。在每个 5 ms 起点，5 ms 任务首先运行，占 1 ms 时间，完成后转入处理被终止的任务。如果这个时刻时间到达 10 ms 的起点，则启动两个 10 ms 任务运行，各占 2 ms 时间(共占 4 ms)，完成后转入处理被终止的任务。以此类推，确保在 1 s 周期内所有任务按规定的时间序列执行，如图 1-21 所示。

暂不考虑操作系统的时间开销时，系统可被正常调度。

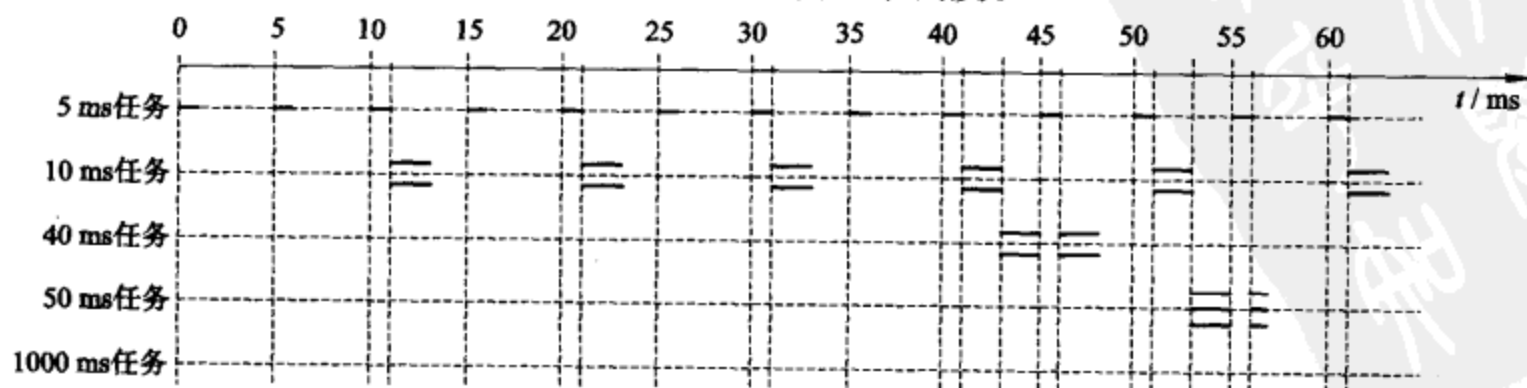


图 1-21 各周期任务的工作时序示意图

1.3.4 2006 年下半年嵌入式系统设计师下午试卷试题 1

一、试题描述

阅读下面有关车载 GPS 终端系统的叙述，回答问题 1 至问题 3，并将答案填入答题纸的对应栏中。(15 分)

【说明】

GPS 就是全球定位系统(Global Position System)。车载 GPS 终端是置于机动车内的实时定位装置，它的应用对象是需要定位和调度的车辆。车辆可以通过终端与 GPS 进行实时、准确的定位，并能够通过无线通信网络上报远程车辆调度中心。中心可以通过终端远程监视车行轨迹，并可在特殊情况下通过终端控制车辆。同时，终端还装备车载电话，可以在出现特殊情况时及时通知车辆调度中心。

图 1-22 所示为车载终端系统的用例图，对于车载 GPS 终端系统来说，主要是车辆调度中心用户和车载终端用户。

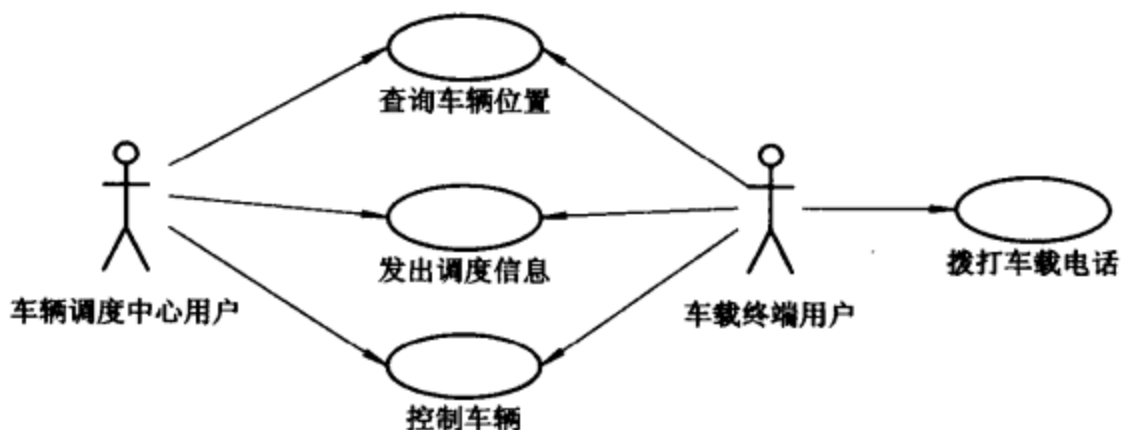


图 1-22 车载终端系统用例图

图 1-23 所示为车载终端系统中的 GSM 无线通信模块的状态图，用于与调度中心进行联系。GSM 模块共有四个状态，分别是通话中、有问题、待命和短消息通信中。

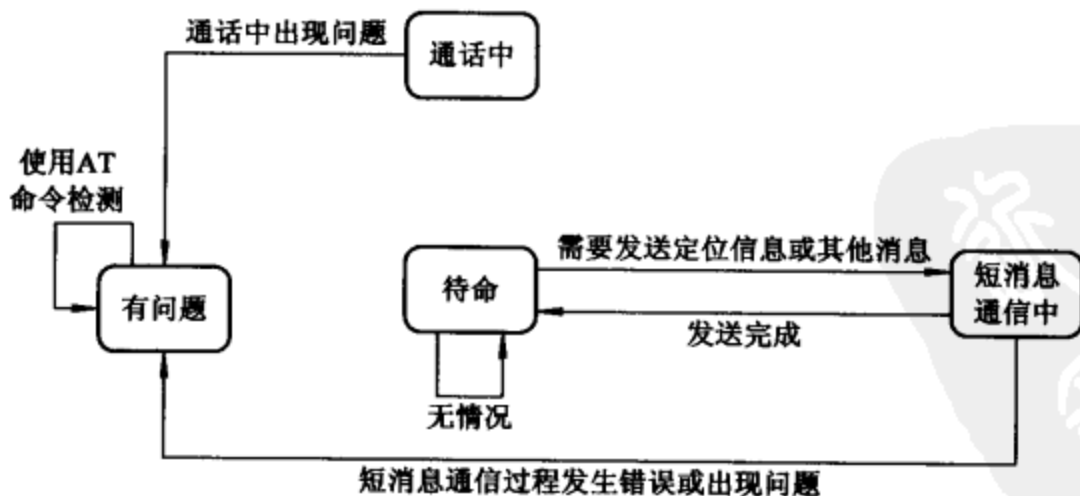


图 1-23 无线通信模块状态图

图 1-24 为车载终端用户在遇到特殊情况下通过车载电话或按键与调度中心保持通信的处理过程顺序图。

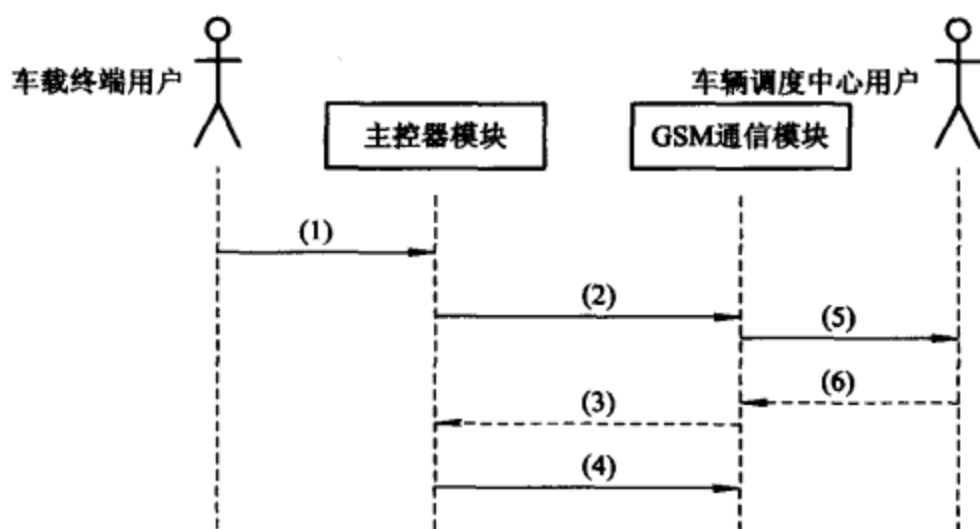


图 1-24 处理过程顺序图

【问题 1】(4 分)

图 1-23 的状态图不完全, 请将其补充完整, 并将下面给出的转换关系填入图 1-23 的适当位置。

转换关系: ① 用户需要语言通话

② 通话完成

③ 重新连接网络

④ 未找到网络或网络出错

【问题 2】(6 分)

按照你对 GPS 终端系统的理解, 将下列 A~F 标记的处理过程填入到图 1-24 中的 (1)~(6) 处, 并选择正确的处理顺序。

A. 语音对话/按下按钮

B. 发送信息到通信模块

C. 监听命令

D. 接收成功要求监听信息

E. 要求监听

F. 通过 GSM 发送信息

处理顺序选项:

① ABDFCE

② ACDBFE

③ ABCDEF

④ ACBDEF

【问题 3】(5 分)

画出 UML 模型中常用关系符依赖、泛化、关联、聚合和实现的图示。

二、要点解析**【问题 1】(4 分)**

统一建模语言(UML)是面向对象的建模语言, 强调两个重要概念: 鼓励将设计描述为许多交互的对象, 而不是一些大的单块代码; 至少一些对象对应系统中部分实际的软件或硬件, 可以用 UML 模型实现系统与外部世界的交互, 在这种情况下, 对象可能与人或其他机器对应。

图 1-25 是车载终端系统中的 GSM 无线电通信模块的状态图, GSM 模块共有 4 个状态: 通话中、有问题、待命和短消息通信中。当 GSM 模块在通话、待命和短消息状态中出现问题时, 会转入错误处理即进入问题状态。当用户需要语音通话时, 转入通话状态, 通话完毕后, 通信模块重新回到待命状态。当模块无法处理问题时, 可以试图连接网络, 上报车辆调度中心, 此时模块处于空闲待命状态。

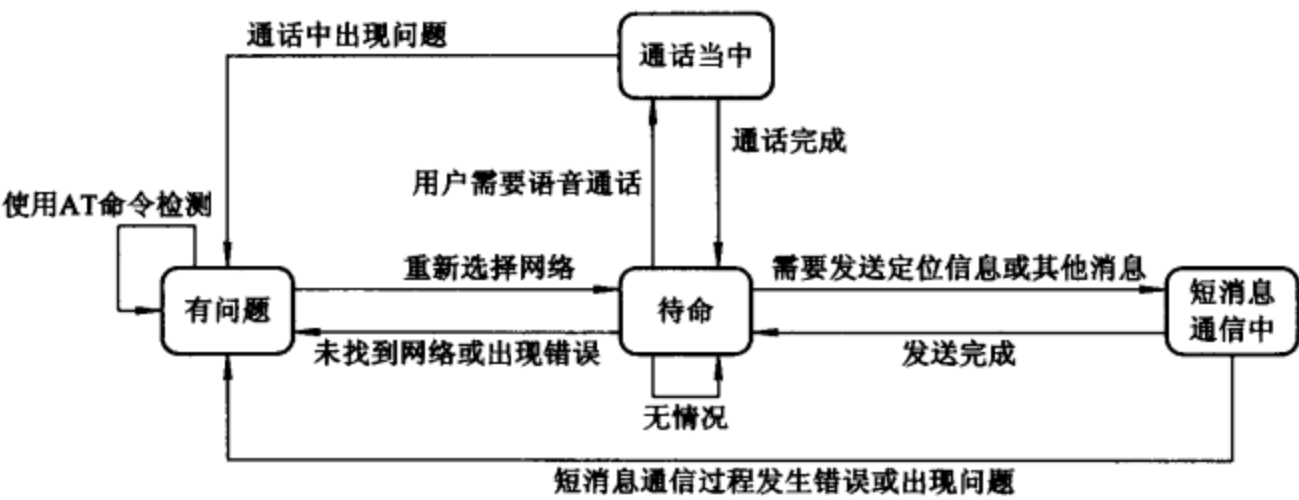


图 1-25 无线通信模块状态图

【问题 2】(6 分)

图 1-25 为车载终端用户通过车载电话或按键与调度中心保持通信的处理过程顺序图。其表达的处理过程如下：用户通过按下按钮或语音对话试图连接调度中心，主控制器模块接到连接请求后连接建立后可以发送信息给通信模块，通信模块负责处理并通过 GSM 发送消息，调度中心接收消息成功后要求监听信息，并把该要求返回给车载系统的 GSM 通信模块，GSM 通信模块把该请求信息（即调度中心的要求监听信息）交给主模块，主模块把监听命令交给 GSM 通信模块，然后可由 GSM 通信模块交给调度中心。

- (1) A，或语音对话/按下按钮(0.5 分)
- (2) B，或发送信息到通信模块(0.5 分)
- (3) E，或要求监听(0.5 分)
- (4) C，或监听命令(0.5 分)
- (5) F，或通过 GSM 发送信息(0.5 分)
- (6) D，或接收成功要求监听信息(0.5 分)

正确的处理顺序为：③ 或 ABCDEF(3 分)

【问题 3】(5 分)

请参见图 1-26。

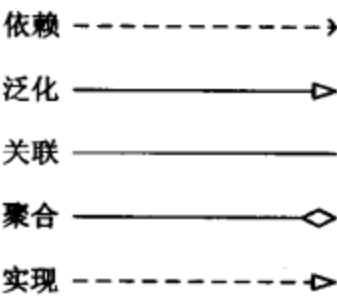


图 1-26 UML 模型中常用关系符图示

第2章 接口电路设计及其编程

2.1 学习要点

2.1.1 考纲要求

根据下午考试大纲中相应的考核要求,在嵌入式系统接口电路的软硬件设计方面,要求考生主要掌握以下几方面的内容。

一、嵌入式系统硬件基本结构

1. 嵌入式微处理结构与应用
2. 异常及中断处理技术
3. DMA 技术
4. 多处理系统
 - (1) 多处理器系统特点;
 - (2) 多处理器系统构建技术。
5. 总线架构

应用系统中的总线配置。

6. 内存种类及架构
存储器系统接口设计。
7. 数字电路和逻辑电路
 - (1) 专用集成电路;
 - (2) 可编程逻辑控制器件。

二、输入/输出接口设计

1. 输入/输出接口
 - (1) 接口信号电平转换;
 - (2) 接口驱动电路设计。
2. 输入/输出接口应用技术
 - (1) 外围设备;
 - (2) 串口通信;
 - (3) 并口通信;
 - (4) 模拟接口;
 - (5) 通信接口设备;

- (6) 通信标准和协议;
- (7) 数据传输方式。

三、外围设备接口应用技术

1. 外围存储设备

存储卡、记忆棒、IC 卡、MMC 卡、SD 卡、DVD、CD-R、CD-RW。

2. 外围输入/输出设备

- (1) 键盘、鼠标、触摸屏;
- (2) 液晶板、LED、7 段数码管、蜂鸣器。

3. 电源设计技术

四、可靠性与安全性设计技术

1. 错误检测与隔离技术

2. 冗余设计

3. 系统恢复设计

4. 诊断技术

5. 常用安全标准

6. 抗干扰设计

7. 电磁兼容设计

8. 系统加密

五、硬件设计环境

1. 硬件描述语言

- (1) 硬件开发设计过程;
- (2) 硬件描述语言的种类与特点。

2. 仿真技术

(1) 逻辑仿真方法;

(2) 逻辑仿真工具。

3. 大规模集成电路系统的开发方法

(1) ASIC 开发方法;

(2) FPGA 设计方法;

(3) IP(Intellectual Property)。

六、嵌入式系统低功耗设计技术

1. 低功耗系统工作机制

2. 低功耗系统模型结构

3. 低功耗的硬件设计技术

2.1.2 学习建议

- 做为一名合格的嵌入式系统设计师,应当熟练掌握嵌入式系统产品的各种接口电路的

软硬件设计方法。从2006年下半年嵌入式系统设计师考试下午试卷的内容来看,本章节知识点考核要求主要是结合实际应用环境考查 RS-232 异步串口接口电路设计,该部分考试难度中等。由于嵌入式系统接口电路涉及的内容比较广,所考查的知识点也比较细腻,因此这一部分试题的命题范围将越来越窄,所考查的知识点也越来越细,试题难度(主要体现在深度方面)也会随之增大。这一部分试题今后可能的一个出题方向是各种接口电路的软件初始化程序的设计。为了解答这类试题,考生除了熟悉有关理论、方法和配置步骤之外,还需要具有丰富的实践配置经验,并善于应用归纳和对比等学习方法进行知识点的学习。

建议考生一定要在这方面的知识点上多下功夫,尽可能地创造条件在开发板(或实验板)上熟悉每一种接口电路的设计,并读懂相关的初始化配置信息。考生可以多下心思挖掘接口电路的考核知识点,努力做到熟能生巧,以便考试时能灵活变通,节约在这方面知识点的解题思考时间。本章也将力求以发展的眼光和实用的角度来预测网络新技术等相关考核点,以增强考生学习相关知识点的目的性。

通过本章学习,可获得以下理论知识点及工程实践经验:

- ① 嵌入式系统外接 Flash Memory 技术;
- ② 嵌入式系统键盘接口电路;
- ③ 嵌入式系统 LCD 接口电路设计;
- ④ 嵌入式系统串口通信技术;
- ⑤ 嵌入式系统网络接口技术。

2.2 典型案例

2.2.1 案例1 外接 Flash Memory 技术

一、案例描述

阅读以下关于嵌入式系统外接 Flash Memory 的技术说明,结合电路原理图回答问题1至问题5。(15分)

【说明】

Flash Memory 是嵌入式系统的重要组成部分,它可用来存储程序和数据,而且可以在掉电情况下继续保存数据而使其不会丢失。某嵌入式系统的 BOOT FLASH 二连接电路如图2-1所示。其中,S3C44BOX 芯片是 SAMSUNG 公司推出的 16/32 位 RISC 处理器,它使用 ARM7TDMI 内核,提供了丰富的内置部件,包括 8 KB Cache 和内部 SRAM、LCD 控制器、带自动握手的二通道 UART、四通道 DMA、系统管理器、带 PWM 功能的五通道定时器和一个内部定时器、I/O 端口、RTC、八通道 10 位 ADC、II C-BUS 接口、II S-BUS 接口、同步 SIO 接口、PLL 倍频器等。S3C44BOX 的体系结构具有如下特性:

- ① 16/32 位 RISC 体系结构和 ARM7TDMI 处理器(66 MHz)内核强大的指令体系;
- ② Thumb 代码压缩机,最大化代码密度同时保持了 32 位指令的性能;
- ③ 一个基于 JTAG 的片上集成 ICE 断点调试方式;

- ④ 一个 32×8 位硬件乘法器；
- ⑤ 实现低功耗 SAMBA II (三星 ARM CPU 嵌入式微处理器总线结构) 的新型总线结构。
- ⑥ 集成了手持设备和通用嵌入式系统应用的解决方案。

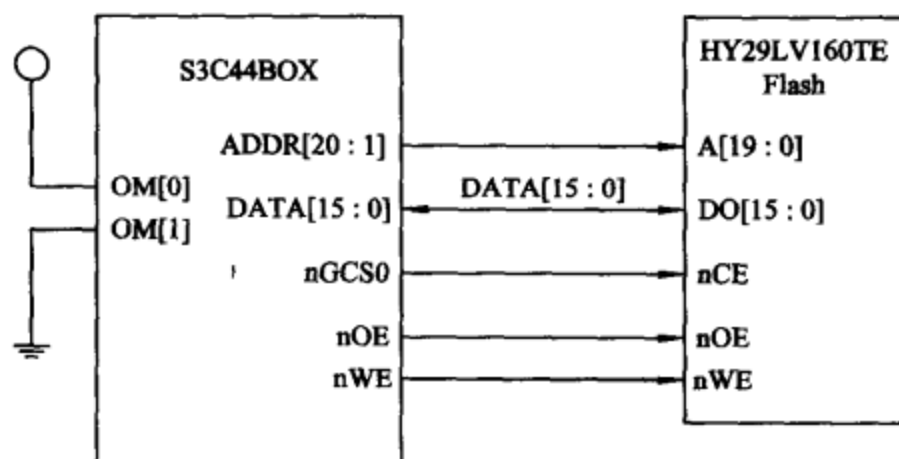


图 2-1 BOOT FLASH 二连接电路

S3C44BOX 的存储系统具有以下一些主要特性：

- ① 可通过外部引脚进行大、小端数据存储模式选择；
- ② 地址空间具有 8 个存储体，每个存储体的容量可达 32 MB，总共可达到 256 MB；
- ③ 对所有存储体的访问大小均可进行改变 (8 位/16 位/32 位)，对总线宽度可进行编程；
- ④ 8 个存储器中，Bank0~Bank5 可支持 ROM、SRAM，Bank6、Bank7 可支持 ROM、SRAM 和 FP/EDO/SDRAM 等，Bank6 和 Bank7 存储容量大小相同；
- ⑤ 7 个存储体具有固定的存储体起始地址和可编程存储体大小；一个存储体具有可编程的存储体起始地址和存储体大小 (即第 8 个可变)；
- ⑥ 所有的存储器 Bank 具有可编程的操作周期；
- ⑦ 支持外部等待信号延长总线周期；
- ⑧ 支持掉电时 DRAM/SDRAM 的自刷新模式；
- ⑨ 支持均匀/非均匀的 DRAM 地址。

图 2-1 中 HY29LV160TE 芯片是 HYUNDAI 公司生产的 16 位 Flash 存储器，其主要特点如下：

- ① 3 V 单电源供电，内部可产生高电压进行编程和擦除操作；
- ② 支持 JEDEC 单电源 Flash 存储器标准和 CFMI (Common Flash Memory Interface) 特性；
- ③ 只需向其命令寄存器写入标准的微处理器指令，具体编程、擦除操作由内部嵌入的算法实现，并且可以通过查询特定的引脚或数据线监控操作是否完成；
- ④ 可以对任一扇区进行读、写或擦除操作，而不影响其他部分的数据。

【问题 1】(5 分)

Flash Memory 是一种非易失存储器，根据结构的不同可分为 NOR Flash 和 NAND Flash 两种。在选择存储解决方案时，应从哪几个方面权衡这两种不同结构 Flash Memory 的选择？

【问题 2】(2 分)

对于图 2-1 的电路连接,请指出 HY29LV160TE 芯片的可用地址范围。

【问题 3】(3 分)

HY29LV160TE 操作编程命令包括读、擦除、编程和复位等命令,如表 2-1 所示。

表 2-1 HY29LV160TE 部分编程命令操作流程

命令序列		总步数	第 1 步		第 2 步		第 3 步		第 4 步		第 5 步		第 6 步	
			地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据
读/复位	字	3	555H	AAH	2AAH	55H	555H	F0H	RA	RD	—	—	—	—
	字节		AAAH		555H		AAAH							
自动选择	字	3	555H	AAH	2AAH	55H	555H	90H	—	—	—	—	—	—
	字节		AAAH		555H		AAAH							
编程	字	4	555H	AAH	2AAH	55H	555H	A0H	RA	PD	—	—	—	—
	字节		AAAH		555H		AAAH							
片擦除	字	6	555H	AAH	2AAH	55H	555H	80H	555H	AAH	2AAH	55H	555H	10H
	字节		AAAH		555H		AAAH		AAAH		555H		AAAH	
扇区擦除	字	6	555H	AAH	2AAH	55H	555H	80H	555H	AAH	2AAH	55H	555H	30H
	字节		AAAH		555H		AAAH		AAAH		555H		AAAH	

HY29LV160TE 芯片具有三种擦除方式,分别是整片擦除、扇区擦除和快擦除。参考表 2-1 扇区擦除的操作流程,将以下用 C 语言代码编写的扇区擦除子程序中的(1)~(3)空缺处填写完整。

```
void F29LV160_EraseSector(int targetAddr)
{
    Uart_Printf("Sector Erase is started! \n");
    _RESET( );           //芯片复位
    _WR(0x555, 0xaa);
    _WR((1));
    _WR((2));
    _WR(0x555, 0xaa);
    _WR(0x2aa, 0x55);
    _WR(BADDR2WADDR(targetAddr), (3));
    _WAIT( );           //等待擦除结束
    _RESET( );           //芯片复位
}
```

其中 _WR(addr, data) 是写入一个字函数:

```
#define _WR(addr, data) * ((U16 *) (addr << 1)) = (U16) data
#define BADDR2WADDR(Addr) >> 1
```

【问题 4】(3 分)

HY29LV160TE 芯片进行整片擦除工作之后,就能够在 Flash 的任意地址中进行写入操作了。参考表 2-1 操作流程及 _WR(addr, data) 函数,得到的字写入子程序,并将子程序中的(4)~(6)空缺处填写完整。


```

int F29LV160_ProgFlash(U32 realAddr, U16 data)
{
    volatile U16 * tempPt;
    int temp, count=0;
    tempPt=(volatile U16 *)realAddr;
    _WR(0x555, 0xaa);
    (4);
    (5);
    * tempPt=data;
    return _WAIT( );
}

int _WAIT(void)                                //检测 DQ6 输出是否停止翻转
{
    volatile U16 flashStatus, old;
    old=*((volatile U16 *)0x0);
    while(1)
    {
        flashStatus=*((volatile U16 *)0x0);
        if((old&0x40) == (flashStatus&0x40))
            break;
        if(flashStatus&0x20)                    //测试超时标志位 DQ5
        {
            old=*((volatile U16 *)0x0);
            flashStatus=*((volatile U16 *)0x0);
            if((old&0x40) == (flashStatus&0x40))
                return 1;                        //无翻转
            else
                (6)
        }
        old=flashStatus;
    }
    return 1;
}

```

【问题 5】(2 分)

在图 2-1 所示的电路图中,若定义了一个 int 类型的常量 $a=0x8192F3A5$,如果按大端存储法存放于 $0x6000$ 内存地址处,那么请将图 2-2 中的(7)~(10)空缺处填写完整,以表达出这种存放顺序。

6000	6001	6002	6003
(7)	(8)	(9)	(10)

图 2-2 大端存储法存放顺序图

二、要点解析

【问题 1】(5 分)

这是一道要求读者掌握选择嵌入式系统存储器需考虑哪些因素的简答题。本题所涉及的知识点有：

在选择存储解决方案时，应从表 2-2 中的七个方面考虑 NOR Flash 和 NAND Flash 存储器的选择。

表 2-2 NOR Flash 和 NAND Flash 存储器选型比较表

	NOR Flash	NAND Flash
① 器件性能	读速度比 NAND Flash 稍快一些	写入速度、擦除速度比 NOR Flash 快，但随机读取能力差，适合大量数据的连续读取
② 接口差别	带有 SRAM 接口，有足够的地址引脚来寻址，很容易存取内部的每一个字节	地址、数据和命令共用 8 位总线，每次读/写采用 512 B 的块，且操作规程较复杂
③ 容量和成本	容量较小，通常在 1~8 MB 之间	只用在 8 MB 以上的产品，单位存储量的价格较低
④ 可靠性和耐用性	擦写次数 10 万次左右；较少出现位交换现象	每个块擦写次数 100 万次左右，具有 10:1 的块擦除周期优势；可能会出现位交换现象；坏块是随机分布的
⑤ 易用性	可直接使用	需要相应的 I/O 接口支持
⑥ 软件支持	运行代码时不需要任何的软件支持；写入和擦除操作需要 MTD(内存技术驱动程序)	运行代码、写入和擦除都需要内存技术驱动程序(MTD)
⑦ 市场定位	代码闪存，用于对数据可靠性要求较高的领域	数据闪存，用于存储容量要求较高的领域

【问题 2】(2 分)

这是一道要求读者掌握嵌入式系统存储器地址空间计算方法的试题。本题的解答思路是，在图 2-1 的电路连接中，S3C44BOX 处理器通过片选 nGCS0 与片外 HY29LV160TE 芯片连接。由于 HY29LV160TE 芯片是 16 位的 FLASH，因此应该用 CPU 的地址线 A1~A20 来分别与 FLASH 的地址线 A0~A19 连接。HY29LV160TE 芯片的可用地址空间范围是 0x00000000~0x001FFFFFFF。

【问题 3】(3 分)

这是一道要求读者编写嵌入式系统存储器擦除子程序的编程题。本题的解答思路如下：

① 注意题目中的关键信息“扇区擦除”。依据此信息查阅表 2-1 中相应的扇区擦除编程命令的操作流程。

② 对于写入一个字的 `_WR(addr, data)` 函数, 其中“`#define _WR(addr, data) * ((U16 *) (addr << 1)) = (U16) data`”说明 `addr` 为写入数据的目标地址, 由于 `data` 为 16 位字而不是字节, 因此 `addr` 也是字地址。“`#define BADDR2WADDR(Addr) >> 1`”说明 `addr` 必须加以调整, 即左移 1 位, 这是因为通常嵌入式微处理器的地址是以字节为单位的, 所以需做此调整。

③ 通过以上分析, 可从表 2-1 中扇区擦除编程命令的操作流程中整理出与本试题相关的内容, 见表 2-3。

表 2-3 HY29LV160TE 扇区擦除命令字操作流程

命令序列		总步数	第 1 步		第 2 步		第 3 步		第 4 步		第 5 步		第 6 步	
			地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据
扇区擦除	字	6	555H	AAH	2AAH	55H	555H	80H	555H	AAH	2AAH	55H	555H	30H

根据表 2-3 的操作流程, 对题目程序段中已给出的命令行进行分析:

```

_WR(0x555, 0xaa);           //扇区擦除流程的第 1 步
_WR(0x2aa, 0x55);           //扇区擦除流程的第 2 步
_WR(0x555, 0x80);           //扇区擦除流程的第 3 步
_WR(0x555, 0xaa);           //扇区擦除流程的第 4 步
_WR(0x2aa, 0x55);           //扇区擦除流程的第 5 步
_WR(BADDR2WADDR(targetAddr), 0x30); //扇区擦除流程的第 6 步

```

由此可得, (1) 空缺处应填入“0x2aa, 0x55”; (2) 空缺处应填入“0x555, 0x80”; (3) 空缺处应填入“0x30”。

【问题 4】(3 分)

这是一道要求读者编写嵌入式系统存储器字写入子程序的编程题。本题的解答思路可借鉴问题 3 的分析思路, 具体解答过程如下:

① 注意题目中的关键信息“字写入”, 依据此信息查阅表 2-1 中相应的字写入编程命令的操作流程, 并整理出与本试题相关的内容, 见表 2-4。

表 2-4 HY29LV160TE 字写入编程命令操作流程

命令序列	总步数	第 1 步		第 2 步		第 3 步		第 4 步		第 5 步		第 6 步	
		地址	数据	地址	数据	地址	数据	地址	数据	地址	数据	地址	数据
编程	字	4	555H	AAH	2AAH	55H	555H	A0H	RA	PD	—	—	—

② 根据表 2-4 的操作流程, 结合问题 3 扇区擦除的编程思路, 对题目程序段中已给出的命令行进行分析:

```

_WR(0x555, 0xaa);           //字写入编程流程的第 1 步
_WR(0x2aa, 0x55);           //字写入编程流程的第 2 步
_WR(0x555, 0xa0);           //字写入编程流程的第 3 步

```

由此可得, (4) 空缺处应填入“`_WR(0x2aa, 0x55)`”; (5) 空缺处应填入“`_WR(0x555, 0xa0)`”。

③ 在字写入操作的最后,有一个判断写入是否完成的`_WAIT()`的子程序。在`_WAIT()`子程序中,采用 Toggle bit(翻转位)的方法检测 DQ6 脚的输出,若翻转停止则表示字写入完成。

④ 由题目程序段中条件判断“`if((old&0x40) == (flashStatus&0x40))return 1;`”,当条件满足时(即 Toggle bit 无翻转),则返回 1;而对于 else 条件不满足的程序段(即 Toggle bit 翻转时,表明字写入完成),则返回 0。由此可得,(6)空缺处应填入“`return 0`”。

【问题 5】(2 分)

这是一道要求读者掌握嵌入式系统存储器大/小端存储法的分析题。本题的解答思路如下:

① 在图 2-1 电路图中,S3C44BOX 芯片是一块 32 位(4 字节)RISC 处理器。

② 大端字节顺序存储法是将高字节数据存放在低地址处,低字节数据存放在高地址处,如表 2-5 第二行所示。

③ 小端字节顺序存储法是将低字节数据存放在低地址处,高字节数据存放在高地址处,如表 2-5 第三行所示。

表 2-5 大/小端存储法示意表

地址 存储方式	6000	6001	6002	6003
大端存储法	81	92	F3	A5
小端存储法	A5	F3	92	81

三、参考答案

表 2-6 给出了本案例问题 1 至问题 5 的参考答案,供读者练习时参考,以便查缺补漏。读者也可依照所给出的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

表 2-6 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (5 分)	① 器件性能 ② 接口差别 ③ 容量和成本 ④ 可靠性和耐用性 ⑤ 易用性 ⑥ 软件支持 ⑦ 市场定位(答出其中 5 点即可,每小点 1 分)	
【问题 2】 (2 分)	0x00000000~0x001FFFFFFF (2 分)	
【问题 3】 (3 分)	(1) 0x2aa, 0x55 (2) 0x555, 0x80 (3) 0x30 (每空 1 分)	
【问题 4】 (3 分)	(4) <code>_WR(0x2aa, 0x55)</code> (5) <code>_WR(0x555, 0xa0)</code> (6) <code>return 0</code> (每空 1 分)	
【问题 5】 (2 分)	(7) 81 (8) 92 (9) F3 (10) A5 (每空 1 分)	

2.2.2 案例2 键盘接口电路

一、案例描述

阅读以下关于嵌入式系统键盘驱动技术的说明,结合电路原理图回答问题1至问题4。
(15分)

【说明】

S3C44BOX 具有 71 个多功能输入/输出脚。它们包含在 7 组端口中:两个 9 位输入/输出端口(端口 E 和 F),两个 8 位输入/输出端口(端口 D 和 G),一个 16 位输入/输出端口(端口 C),一个 10 位输出端口(端口 A),一个 11 位输出端口(端口 B),用它们可以组成各种各样的电路。

每组端口都可以通过软件配置寄存器来满足不同系统的设计需要,在运行程序之前必须先对每一个用到的引脚的功能进行设置。如果某些引脚的复用功能没有使用,那么可以将该引脚设置成 I/O 口。S3C44BOX 芯片中有端口配置寄存器(PCONA~G)、端口数据寄存器(PDATA~G)、端口上拉设置寄存器(PUPC~G)、外部中断控制寄存器等端口控制寄存器。

由于这 7 组端口中多数都是多功能口,因此需要用“端口配置寄存器 PCONn”来设置每个引脚工作时的功能模式。

当端口被设置为输出脚时,输出数据的方法就是将数据写入到端口数据寄存器 PDATn 的相应位中;当端口被设置为输入脚时,读入数据的方法就是将 PDATn 中的相应位读出。

端口上拉寄存器(PUPC~G)用来设定 PC、PG 这几组端口是否具有内部上拉功能。当 PUPn 的对应位为 0 时,该引脚上的上拉功能将使能;当为 1 时,该引脚上的上拉功能将不起作用。

外部中断控制寄存器是为 PG 的第三功能(外部中断输入口功能)设置的,该寄存器可用来设置 EXINT(外部中断)请求输入的模式,分别有低电平触发、高电平触发、下降沿触发、上升沿触发和边沿触发。

图 2-3 利用 S3C44BOX 的 I/O 口及外部中断构成一个 4×4 行列式矩阵键盘扫描电路。该键盘采用中断扫描方式进行工作,行线输出选用 S3C44BOX 的 GPG4~GPG7,列线输入选用 S3C44BOX 的 GPF5~GPF8。各条行线接上拉电阻保持高电平,并通过与门将输出信号与微处理器的中断 EXINT0 连接;各条列线接上拉电阻保持高电平。

表 2-7 和表 2-8 分别列出了 GPF 和 GPG 端口的引脚各功能定义;表 2-9 给出了端口配置寄存器 PCONF、PCONG 的相关定义。

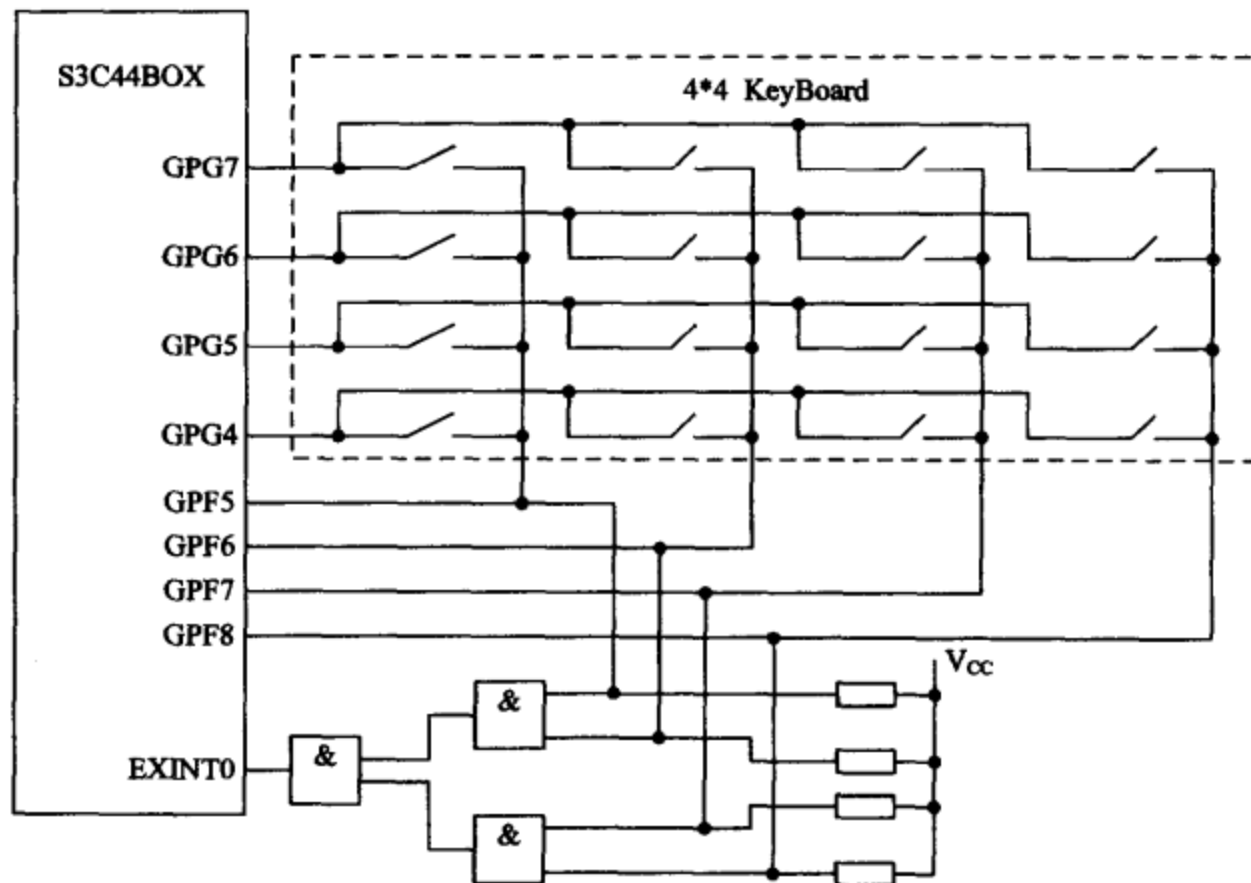


图 2-3 键盘接口电路原理图

表 2-7 GPF 端口各引脚功能定义表

端口 F	可选择的引脚功能			
	功能 1	功能 2	功能 3	功能 4
GPF0	输入/输出	IIC_SCL	—	—
GPF1	输入/输出	IIC_SDA	—	—
GPF2	输入/输出	nWAIT	—	—
GPF3	输入/输出	nXBACK	nXDACK0	—
GPF4	输入/输出	nXBREQ	nXDREQ0	—
GPF5	输入/输出	nRTSL	SIOTxD	IISLRCK
GPF6	输入/输出	TxD1	SIORDY	IISDO
GPF7	输入/输出	RxD1	SIORxD	IISDI
GPF8	输入/输出	nCTSL	SIOCLK	IISCLK

表 2-8 GPG 端口各引脚功能定义表

端口 G	可选择的引脚功能		
	功能 1	功能 2	功能 3
GPG0	输入/输出	VD4	EINT0
GPG1	输入/输出	VD5	EINT1
GPG2	输入/输出	nCTS0	EINT2
GPG3	输入/输出	nCTS0	EINT3
GPG4	输入/输出	IISCLK	EINT4
GPG5	输入/输出	IISDI	EINT5
GPG6	输入/输出	IISDO	EINT6
GPG7	输入/输出	IISLRCK	EINT7

表 2-9 端口配置寄存器 PCONF、PCONG 的相关定义表

寄存器名称		位数/引脚	端口地址	描 述	复位值
PCONF	[21: 10]	3	0x01D20034	000=输入 001=输出 010=功能 2 011=功能 3 100=功能 4 其他=保留	0x0000
	[9: 0]	2		00=输入 01=输出 10=功能 2 11=功能 3	
PCONG	[15: 0]	2	0x01D20040	00=输入 01=输出 10=功能 2 11=功能 3	0x00

【问题 1】(2 分)

键盘的结构通常有两种形式, 分别是线性键盘和矩阵键盘。矩阵键盘根据识键和译键方法的不同又分为哪几种键盘?

【问题 2】(4 分)

仔细阅读图 2-3 的电路原理图, 结合表 2-7、表 2-8 和表 2-9 完成 S3C44BOX 芯片相应 I/O 端口的初始化, 将以下 I/O 端口初始化程序段(1)~(4)空缺处填写完整。

```

rPDATF=0x60;
rPCONF=(1);
rPUPF=(2);
rPDATG=0x0;
rPCONG=(3);
(4);

```

【问题 3】(4 分)

S3C44BOX 的中断控制器可以接收来自 30 个中断源的请求。中断控制器的作用,就是响应来自 FIQ 或 IRQ 的中断,并请求内核对中断进行处理。当有多个中断同时发生时,中断控制器要决定首先处理哪一个中断。

PSR 指 ARM7TDMI 处理器的程序状态寄存器。如果 PSR 的 F 位被设置为 1,处理器将不接受来自中断控制器的 FIQ。如果 PSR 的 I 位被设置为 1,处理器将不接受来自中断控制器的 IRQ。因此,为了使能中断相应机制,PSR 的 F 位或 I 位必须被清零,同时 INT-MASK 的相应位必须被清零。

中断模式寄存器(INTMOD 寄存器)中的 26 个位对应着每一个中断源。如表 2-10 所示,若 INTMOD 中的响应位设置为 1,则 ARM7TDMI 内核将以 FIQ(快速中断)模式响应此中断。否则,以 IRQ(普通中断)模式响应。

在中断屏蔽寄存器(INTMSK 寄存器)中,除了全局屏蔽位,其余的 26 位依次对应着每个中断源。当 INTMSK 的某个屏蔽位为 1,同时该位对应的中断事件发生了,此时 CPU 是不会对中断请求进行响应的。如果屏蔽位为 0,则 CPU 将对中断请求进行响应。如果全局屏蔽位为 1,则所有的中断请求都不会被响应,但是当中断发生时,相应的登记位仍将被置 1。INTMSK 的位定义如表 2-10 所示。

表 2-10 INTMOD/ INTMSK 位定义表

中断源	对应控制位	中断源	对应控制位	中断源	对应控制位
外部中断 0	[25]	BDMA1 中断	[16]	UART0 接收中断	[7]
外部中断 1	[24]	看门狗中断	[15]	UART1 接收中断	[6]
外部中断 2	[23]	UART0/1 错误中断	[14]	IIC—中断	[5]
外部中断 3	[22]	定时器 0 中断	[13]	SIO—中断	[4]
外部中断 4/5/6/7	[21]	定时器 1 中断	[12]	UART0 发送中断	[3]
RTC 时间片中断	[20]	定时器 2 中断	[11]	UART1 发送中断	[2]
ZDMA0 中断	[19]	定时器 3 中断	[10]	RTC 报警中断	[1]
ZDMA1 中断	[18]	定时器 4 中断	[9]	ADC 转换结束中断	[0]
BDMA0 中断	[17]	定时器 5 中断	[8]		

请结合以上提供的信息将(5)、(6)空缺处填写完整,完成图 2-3 中“EXINT0”的中断初始化程序段。

```

rINTCON=0x5;                //非向量模式, IRQ 允许, FIQ 禁止
rINTMOD=(5);
rINTMSK=(6);
pISR_EINT0=(unsigned)keyboard_ISR;    //指定中断服务程序

```

【问题 4】(5 分)

图 2-4 示意了键盘扫描电路(图 2-3)的程序流程图,图 2-5 是该键盘扫描电路程序流程图对应的 C 语言程序段 keyboard_ISR,请将图 2-4 和图 2-5 中(7)~(11)空缺处填写完整。

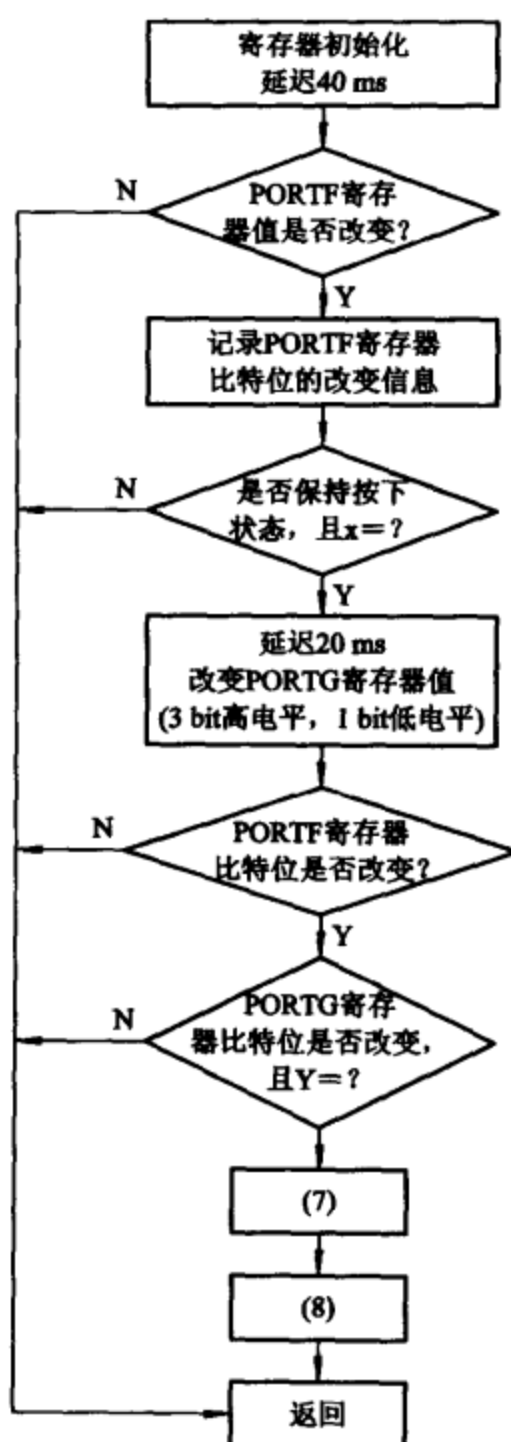


图 2-4 键盘扫描电路程序流程图

```

void keyboard_ISR(void)
{
    char x, y, xrecord, yrecord, temp;
    rI_ISPC=BIT_EINT0; //clear pending_bit
    Delay(400); //delay 40ms
    if((rPDATF&0x1E0)==0x1E0)
    {
        return 0; //no keyboard press, return
    }
    else
    {
        x=1;
        y=1;
    }
}
  
```

```

xrecord=(~((rPDATF&0x1E0)>>1));
xrecord=xrecord>>4;
while(xrecord!=0x1)                                //judge row
{
    x=x+1;
    xrecord=xrecord>>1;
    if(xrecord==0)
    {
        rPDATG=0X0F;                                //no , return
        return 0; }
    }
    Delay(200);                                        //delay 20ms
    (9);                                              //input high
    while((rPDATF&0x1E0)==0x1E0)
    {
        (10);
        temp=rPDATG;
        if((temp&0xf0)==0xf0)                        //noboard press, return
        {
            rPDATG=0x0F;
            return 0;
        }
    }
    Delay(100);                                        //delay 10ms
    yrecord=~((rPDATG&0xf0)>>4)&0x0F;
    while(yrecord!=0x1)                                //judge line
    {
        y=y+1;
        yrecord=yrecord>>1;
        if( (11) )
        {
            rPDATG=0X0F;                                //no board press, return
            return 0;
        }
    }
    Led_Display(x+(y-1)*4-1);
    Uart_Printf("%3d", x+(y-1)*4-1);
    Delay(400);                                        //delay 40ms
    rPDATG=0X0F;
}
}

```

图 2-5 键盘扫描电路程序流程图对应的 C 语言程序段 keyboard_ISR

二、要点解析

【问题 1】(2 分)

这是一道要求读者掌握嵌入式系统人机界面之一键盘基本常识的简答题。本题所涉及的知识点如下:

- ① 键盘的结构通常有两种形式, 分别是线性键盘和矩阵键盘。
- ② 线性键盘由若干个独立的按键组成, 每个按键的一端与微处理器的一个 I/O 端口相连。
- ③ 矩阵键盘的按键按 N 行 M 列排列, 每个按键占据行列的一个交点, 需要的 I/O 端口数目是 $N+M$, 允许的最大按键数是 $N \times M$ 。
- ④ 矩阵键盘根据识键和译键方法的不同又分为非编码键盘和编码键盘两种。
- ⑤ 非编码键盘主要用软件的方法识键和译键。根据扫描方法的不同, 可分为行扫描法、列扫描法和反转法三种。
- ⑥ 编码键盘主要用硬件来实现键的扫描和识别, 通常使用 8279 专用接口芯片, 在硬件上要求较高。

【问题 2】(4 分)

这是一道要求读者掌握嵌入式系统微处理器 I/O 端口初始化的分析题。本题的解答思路如下:

- ① 结合图 2-3 仔细阅读题目所提供的信息得知, 列线 GPF5~GPF8 为输入, 接着查阅表 2-7 得知 GPF 端口的第 5~8 引脚定义为“功能 1 输入/输出”。
- ② 查阅表 2-9 得知, GPF 端口使用一个 21 位端口配置寄存器(PCONF)定义各引脚的功能, 且有使用二位、三位来分别定义每个引脚功能的两种配置方式。可将表 2-7 和表 2-9 中与 GPF 引脚的相关定义组合成表 2-11。

表 2-11 GPF 端口各引脚功能定义表

引脚	PCONF 相对应的位	描 述			
GPF8	[21: 19]	000=输入 011=SI0CLK	001=输出 100=IISCLK	010=nCTSL Others=保留	
GPF7	[18: 16]	000=输入 011=SIORxD	001=输出 100=IISDI	010=RxD1 Others=保留	
GPF6	[15: 13]	000=输入 011=SIORDY	001=输出 100=IISDO	010=TxD1 Others=保留	
GPF5	[12: 10]	000=输入 011=SIOTxD	001=输出 100=IISLRCK	010=nRTSL Others=保留	
GPF4	[9: 8]	00=输入	01=输出	10=nXBREQ	11=nXDREQ0
GPF3	[7: 6]	00=输入	01=输出	10=nXBACK	11=nXDACK0
GPF2	[5: 4]	00=输入	01=输出	10=nWAIT	11=保留
GPF1	[3: 2]	00=输入	01=输出	10=IICSDA	11=保留
GPF0	[1: 0]	00=输入	01=输出	10=IICSCL	11=保留

③ 由表 2-11 中 GPF8~GPF5 的内容可知,要将这几个引脚定义为输入,需将端口配置寄存器(PCONF)的第 10~21 位置 0,因此(1)空缺处可填入“0b0 0000 0000 00xx xxxx xxxx”,其中“x”表示的是 GPF0~GPF4 各引脚在相应开发板中的相关定义。因为本试题中对 GPF0~GPF4 各引脚的功能没有特别的说明,所以可将 PCONF 中“x”部分做清零处理,故(1)空缺处也可填入十六进制数“0x00”。

④ 对于“rPUPF=(2)”语句,查阅题干中相关信息可知,端口上拉寄存器(PUPF)用来设定 PF 端口是否具有内部上拉功能。题目中已给出关键信息“各条列线接上拉电阻保持高电平”,因此需将 PUPF 寄存器中第 5 位~第 8 位置 0,使对应引脚的上拉功能使能,即(2)空缺处可填入“0x0”。

⑤ 可将表 2-8 和表 2-9 中与 GPG 引脚的相关定义组合成表 2-12。

表 2-12 GPG 端口各引脚功能定义表

引脚	PCONG 相对应的位	功能 1		功能 2	功能 3
		00	01	10	11
GPG0	[1: 0]	输入	输出	VD4	EINT0
GPG1	[3: 2]	输入	输出	VD5	EINT1
GPG2	[5: 4]	输入	输出	nCTS0	EINT2
GPG3	[7: 6]	输入	输出	nCTS0	EINT3
GPG4	[9: 8]	输入	输出	IISCLK	EINT4
GPG5	[11: 10]	输入	输出	IISDI	EINT5
GPG6	[13: 12]	输入	输出	IISDO	EINT6
GPG7	[15: 14]	输入	输出	IISLRCK	EINT7

⑥ 因为本键盘采用中断扫描方式进行工作,在电路原理图(图 2-3)中使用了微处理器的中断 EXINT0,查表 2-12 可知,外部中断 0(INT0)对应于 GPG0 引脚。因此 GPG0 所对应的端口配置寄存器(PCONG)[1: 0]位应设置成“11”。

⑦ 由题干信息“行线输出选用 S3C44BOX 的 GPG4~GPG7”,可推理出这些行输出线与端口配置寄存器 PCONG 相关的位是第 8 位~第 15 位,且 GPG4~GPG7 所对应的位应分别设置成“01”。

⑧ 通过以上分析,可知(3)空缺处可填入“0b0101 0101 xxxx xx11”,其中“x”表示的是 GPG1~GPG3 各引脚在相应开发板中的相关定义。因为本试题中对 GPG1~GPG3 各引脚的功能没有特别的说明,所以可将 PCONG 中“x”部分做置 0 处理,所以(3)空缺处可填入十六进制数“0x5503”。

⑨ 题目中已给出关键信息“各条行线接上拉电阻保持高电平”,因此需将 PUPG 寄存器中第 4 位~第 7 位置 0,使对应引脚的上拉功能使能,即(4)空缺处可填入“rPUPG=0x0”。

以下给出的是 GPF 及 GPG 端口完整的 I/O 口初始化程序。

```

rPDATF=0x60;
rPCONF=0x00;           //列线 PORTF5~8 为输入
rPUPF=0x0;             //列线接上拉电阻
rPDATG=0x0;
rPCONG=0x5503;         //行线 GPG4~GPG7 输出, GPG0 作为 EXINT0 连接
rPUPG=0x0;             //行线接上拉电阻

```

【问题 3】(4 分)

这是一道要求读者掌握嵌入式系统微处理器中断初始化的分析题。本题所涉及的知识如下：

① ARM7TDMI 有 FIQ(快速中断请求)和 IRQ(普通中断请求)两种类型的中断模式。S3C44BOX 的中断控制器可以接收来自 30 个中断源的请求。在 30 个中断源中,有 26 个中断源提供给中断控制器。四个外部中断(EINT4/5/6/7)请求是通过“或”的形式提供为一个中断源送至中断控制器,因此,ISR(中断服务程序)要通过读取 EXTINPND[3:0]寄存器来区别这四个中断源(EINT4/5/6/7)。它们的中断处理程序(ISR)必须在处理结束时将 EXTINPND[3:0]中对应位写 1 来清除该位。两个 UART 错误中断占用一个中断源。

② 如果 PSR(程序状态寄存器)的 F 位被设置为 1,处理器将不接受来自中断控制器的 FIQ。如果 PSR 的 I 位被设置为 1,处理器将不接受来自中断控制器的 IRQ。因此,为了使能中断相应机制,PSR 的 F 位或 I 位必须被清零,同时 INTMASK 的相应位必须被清零。

③ 中断模式寄存器(INTMOD)中的 26 个位对应着每一个中断源,如表 2-10 所示。当某个中断在 INTMOD 中的响应位被设置为 1 时,则 ARM7TDMI 内核将以 FIQ(快速中断)模式响应此中断。否则,以 IRQ(普通中断)模式响应。由于本试题所提供的信息没有特殊的要求,因此可认为 EXINT0 是个普通中断,中断模式寄存器中所对应的第 25 位需设置为 0,即(5)空缺处可填入十六进制数“0x0”。

④ 如果中断屏蔽寄存器(INTMSK)的某一位被置 1,则该位对应的中断响应被禁止。如果 INTMSK 某一位被置 0,则该位对应的中断响应被使能。如果某个中断在 INTMSK 的寄存器中的相应位为 1,而这个中断却又发生了,那么它在中断登记寄存器的登记位仍被置位。如果全局屏蔽位被置 1,那么当中断发生时,中断登记位仍被置位,但是所有的中断都不会得到服务。

由于本试题电路图中使用了中断 EXINT0,因此中断屏蔽寄存器中所对应的第 25 位需设置为 0,使该位对应的 EXINT0 中断响应被使能,即(6)空缺处可填入允许 EXINT0 及所有中断屏蔽位响应使能的“~(BIT_GLOBAL|BIT_EINT0)”。

⑤ 另外,与中断管理相关的还有中断登记寄存器、中断请求寄存器(INTPND)、IRQ 矢量模式寄存器、IRQ/FIQ 中断服务寄存器等。其中,如果将所有的中断源定义为 IRQ 中断,并且同一时刻发生了 10 个中断请求,那么可以通过读取中断登记寄存器来了解哪些中断发生了,并对产生的中断依次进行处理。这里是通过软件的方式决定中断服务的优先级的。

中断登记寄存器中的每一位显示了所对应的某个中断请求是否还未被处理。一旦某个登记位被置位,当 PSR(程序状态寄存器)的 I 标志位或 F 标志位被清零时,中断服务程序

就会启动执行。

因为中断登记寄存器是一个只读寄存器,所以在服务程序中要想清除登记位时,需要采用在 IRQ/FIQ 中断模式下的清除中断服务寄存器(I_ISPC 或 F_ISPC)中相应位写入 1 的方式实现。

⑥ 中断请求寄存器(INTPND)中的 26 个位均分别对应着一个中断源,如表 2-10 所示。当中断发生时,中断请求寄存器(INTPND)中相应的登记位就会置 1,说明该中断还未被处理。中断服务程序中必须清除该登记位,从而使系统能够及时响应下一次中断。在多个中断同时发生时,INTPND 将所有发生的中断登记位都置 1。虽然中断请求可以通过 INTMSK 寄存器屏蔽,但是如果被屏蔽的中断发生了,INTPND 中的登记位仍然会被置 1。

中断请求寄存器(INTPND)也是一个只读寄存器,清除登记位的方式是向清除中断服务寄存器(I_ISPC 或 F_ISPC)的相应位写入“1”。

⑦ IRQ 矢量模式寄存器的优先级产生模块包括五个单元,由一个主单元和四个辅单元组成。主单元管理四个辅单元和二个中断源,每个辅单元管理六个中断源。每个辅单元有四个可编程的优先级源 sGn 和二个固定优先级源 sGKn。每个辅单元的四个中断源优先级由 I_PSLV 寄存器决定。另外二个固定中断源的优先级在六个中断源中是最低的。主优先级产生单元通过 I_PMST 寄存器决定四个辅单元、两个中断源之间的优先级。中断源 INT_RTC 和 INT_ADC 在 26 个中断源中优先级是最低的。

如果几个中断请求同时发生,I_ISPR 寄存器则将其中具有最高优先级的中断源对应位置 1。

⑧ I_ISPR/F_ISPR 是 IRQ/FIQ 中断模式下的中断服务寄存器,I_ISPC/F_ISPC 是 IRQ/FIQ 中断模式下的清除中断服务寄存器。

I_ISPC/F_ISPC 清除中断登记位(INTPND),同时也通知中断控制器中断服务(ISR)已经结束。在某个中断的服务(ISR)结束时,该中断相应的登记位也必须被清零。

要将 INTPND 的某一位清零,方法是往 I_ISPC/F_ISPC 的相应位写入 1。在清除 I_ISPC/F_ISPC 时,还必须注意,I_ISPC/F_ISPC 寄存器在 ISR 中只能被操作 1 次。如果没有遵守这两点,在中断请求发生时,在 I_ISPR/F_ISPR 和 INTPND 寄存器中该中断所对应的位可能还是 0。

【问题 4】(5 分)

这是一道要求读者掌握嵌入式系统人机接口程序设计的分析题。本题的解答思路如下:

① 根据图 2-3 和图 2-4 中已给出的信息可大致领会到该键盘扫描程序的基本设计思路是,将行线 PORTG4~PORTG7 的端口输出设为低电平,当有键盘按下时,该行线被拉为低电平,使得 EXINT0 输入也为低电平,MCU 产生中断。中断产生后通过对键盘的行和列进行扫描的方法,可以计算出是哪个键按下,并跳到相应的键盘处理程序中去。为了减少因按键的抖动而使键盘扫描程序做出错误判断,在该键盘扫描程序中增加了必要的延时。

② 根据以上键盘扫描程序的设计思想以及(9)空缺处上一条语句“Delay(200);”,可以定位(9)空缺处所填写的内容与图 2-4 程序流程中“delay 20 ms, change PORTG 3bit hige 1 bit low”相关,因此可将端口数据寄存器 PDATG 的各位由高至低依次设置成

“1110 1111”，即(9)空缺处应填入“ $rPDATG=0XEF$ ”语句。

③ (10)空缺处位于“ $while((rPDATF\&0x1E0)==0x1E0)$ ”循环体之中，该循环体用于完成图 2-4 程序流程中“change the bit until PORTF changed”的判断功能。结合键盘行扫描法的设计思路，判断所读取的作为列输入的 PORTF 数据寄存器中哪位为低电平。在图 2-3 电路中需分四次将做为行输出的 PDATG 数据寄存器相应位置为低电平。由于(9)空缺处端口数据寄存器 PDATG 的初值为“1110 1111”，且 GPG4~GPG7 为行输出线，因此可使用位操作运算符中的左移运算来改变端口数据寄存器 PDATG 相应位的值(GPG4~GPG7)，即(10)空缺处应填入“ $rPDATG=rPDATG\ll 1$ ”语句。该语句的填写也可借鉴“ $while(yrecord!=0x1)$ ”循环体中“ $yrecord=yrecord\gg 1$ ”语句的设计思路。

④ 同理，(11)空缺处位于“ $while(yrecord!=0x1)$ ”循环体之中，该循环体用于完成图 2-4 程序流程中“judge which PORTG is changed $Y=?$ ”的判断功能。借鉴“ $while((rPDATF\&0x1E0)==0x1E0)$ ”循环体中“ $if((temp\&0xf0)==0xf0)$ ”语句的设计思路，以及“ $yrecord=\sim((rPDATG\&0xf0)\gg 4)\&0x0F$ ”语句可推理出，(11)空缺处可填入“ $yrecord==0$ ”语句。

⑤ 仔细阅读图 2-5 键盘扫描的程序段，由“ $Led_Display(x+(y-1)*4-1)$ ”语句可知，keyboard_ISR 程序对某一次按键输入的显示处理按“ $x+(y-1)*4-1$ ”进行，因此(7)空缺处可填入类似“ $keyboard=x+(y-1)*4-1$ ”的内容。

⑥ 同理，由“ $Delay(400)$ ”语句可知，keyboard_ISR 程序每次动态显示的延时时间是 40 ms，因此(8)空缺处可填入类似“ $delay\ 40\ ms\ display\ it$ ”的内容。完整的键盘扫描电路程序流程图如图 2-6 所示。

三、参考答案

表 2-13 给出了本案例问题 1 至问题 4 的参考答案，供读者练习时参考，以便查缺补漏。读者也可依照所给出的评分标准得出自己的测试分数，从而大致评估自己对这一方面知识点的掌握程度。

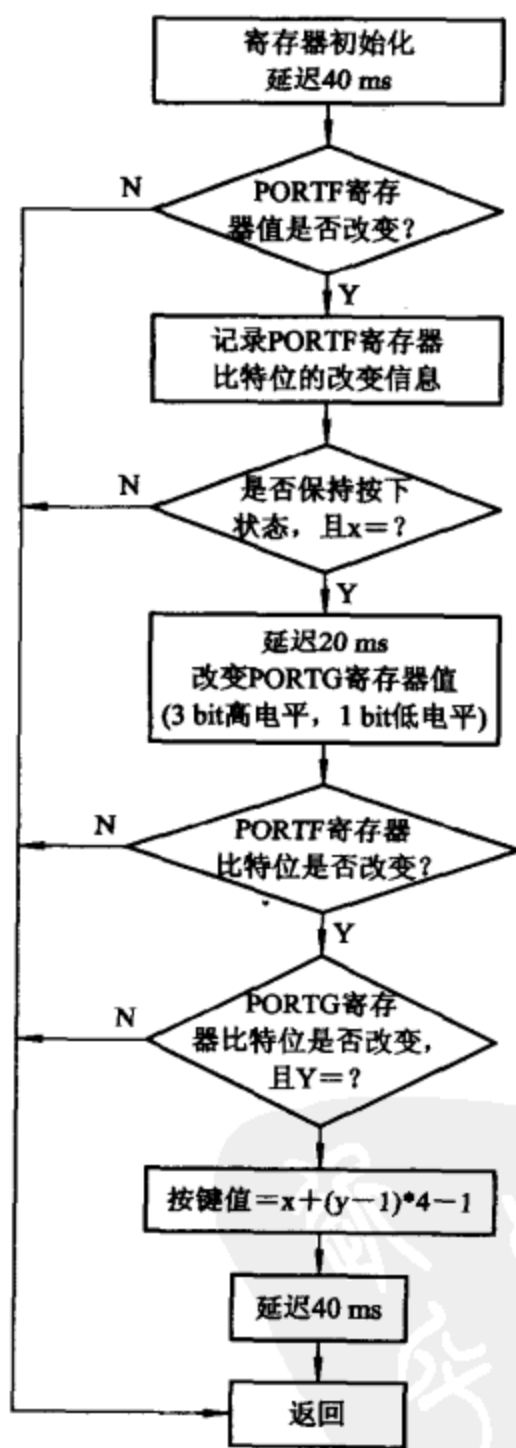


图 2-6 完整的键盘扫描电路程序流程图

表 2-13 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (2 分)	非编码键盘 (1 分) 编码键盘 (1 分)	
【问题 2】 (4 分)	(1) 0x0 (2) 0x0 (3) 0x5503 (4) rPUPG=0x0 (每空 1 分)	
【问题 3】 (4 分)	(5) 0x0 (6) ~(BIT_GLOBAL BIT_EINT0) (每空 2 分)	
【问题 4】 (5 分)	(7) keyboard=x+(y-1)*4-1 (8) delay 40 ms display it (9) rPDATG=0XEF (10) rPDATG=rPDATG<<1 (11) yrecord==0 (每空 1 分)	

2.2.3 案例 3 LCD 接口电路设计及编程技术

一、案例描述

阅读以下关于嵌入式系统 LCD 接口电路设计与编程技术的说明,结合电路原理图回答问题 1 至问题 5。(15 分)

【说明】

LCD 显示器的基本原理是通过给不同的液晶单元供电,控制其光线的通过与否,从而达到显示的目的。LCD 控制器的主要工作是将定位在系统存储器中的显示缓冲区中的 LCD 图像数据传送到外部 LCD 驱动器。

S3C44BOX 中内置的 LCD 控制器,具有将显示缓存(在系统存储器中)中的 LCD 图像数据传输到外部 LCD 驱动电路的逻辑功能。S3C44BOX 内置 LCD 控制器的内部结构框图如图 2-7 所示,它可以驱动规格为每像素 2 位/4 位/8 位的 LCD 屏。对于不同尺寸的 LCD,具有不同数量的垂直和水平像素、数据接口的数据宽度、接口时间及刷新率,而 LCD 控制器可以通过编程来控制相应的寄存器值,以适应不同的 LCD 显示板。

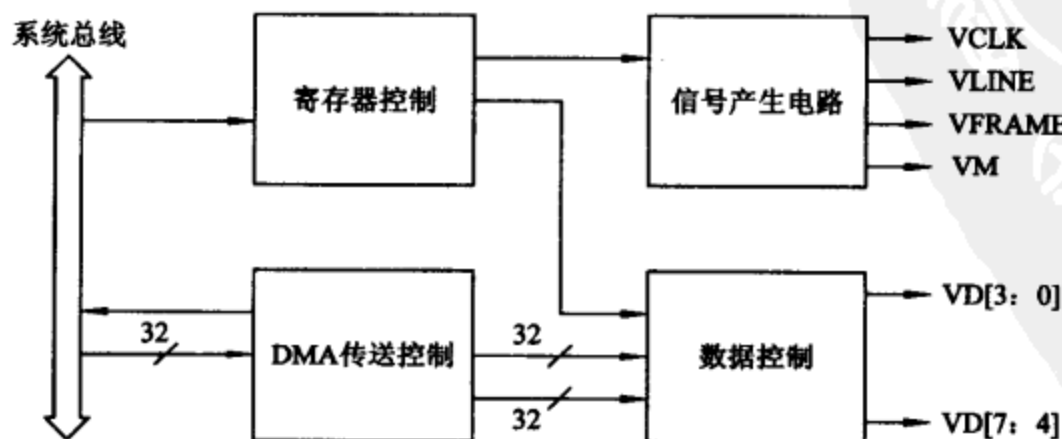


图 2-7 S3C44BOX 内置 LCD 控制器的内部结构框图

图 2-7 中 LCD 控制器的外部接口信号的功能说明见表 2-14。

表 2-14 S3C44BOX 内置 LCD 控制器外接信号功能说明表

接口信号	功 能 说 明
VCLK	此信号为 LCD 控制器和 LCD 驱动器之间的像素时钟信号，由 LCD 控制器送出的数据在 VCLK 的上升沿处送出，在 VCLK 的下降沿处被 LCD 驱动器采样。该信号与 LCD 模块的 XCK 信号相对应
VLIN	LCD 控制器和 LCD 驱动器之间是线同步脉冲信号，此信号用于 LCD 驱动器将水平线（行）移位寄存器的内容传送给 LCD 屏显示。LCD 控制器在整个水平线（整行）数据移入 LCD 驱动器后，插入一个 VLIN 信号，该信号与 LCD 模块的 LP 信号相对应
VFRAME	LCD 控制器和 LCD 驱动器之间是帧同步信号，此信号告诉 LCD 屏新的一帧开始了。LCD 控制器在一个完整帧显示完成后立即插入一个 VFRAME 信号，开始新一帧的显示，该信号与 LCD 模块的 YD 信号相对应
VM	LCD 驱动器所使用的是交流信号，此信号用于改变行和列的电压极性，从而控制像素点的显示或熄灭。该信号可以与每个帧同步，也可以与可变数量的 VLIN 信号同步。该信号与 LCD 模块的 DISP 信号相对应
VD[3:0]	LCD 像素点数据输出端口与 LCD 模块的 D[3:0]相对应
VD[7:4]	LCD 像素点数据输出端口与 LCD 模块的 D[7:4]相对应

图 2-8 给出了一种 S3C44BOX 芯片外接 LCD 模块的驱动电路原理图。

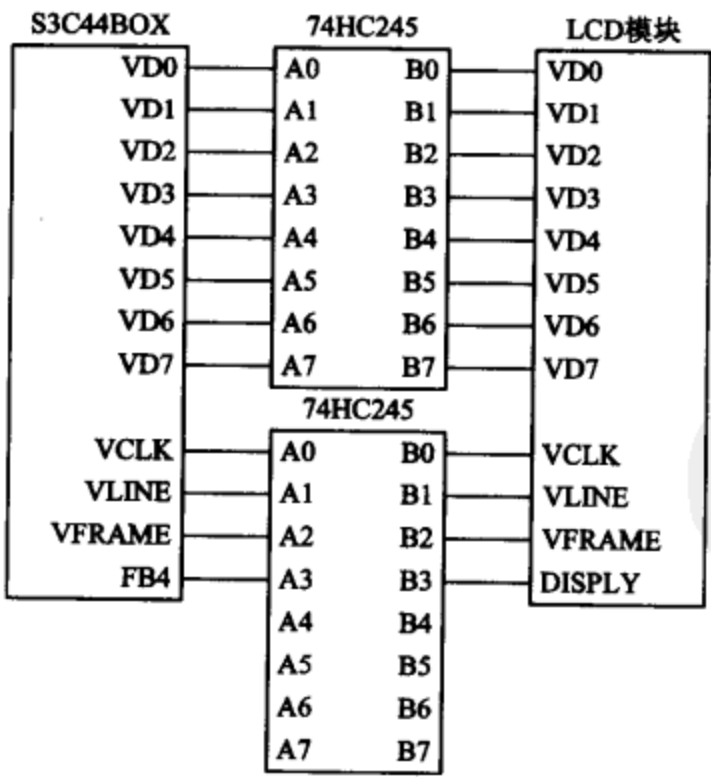


图 2-8 S3C44BOX 芯片外接 LCD 模块的驱动电路原理图

【问题 1】(3 分)

液晶显示所用的液晶材料是一种兼有液态和固态双重性质的有机物，它的棒状结构在

液晶盒内一般平行排列,但在电场作用下其排列方向会改变。液晶显示是一种被动的显示,它本身不能发光,只能使用周围环境的光,但它显示图案或字符只需很小能量。通常 LCD 有哪几种显示方式?

【问题 2】(4 分)

S3C44BOX 的 LCD 控制器支持三种 LCD 驱动器,分别是 4 位双扫描、4 位单扫描和 8 位单扫描数据显示模式。LCD 在 160×240 的单色显示屏中采用 4 位双扫描数据显示模式, LCD 显示缓冲区中的地址与数据如表 2-15 所示,请画出显示缓冲区与 LCD 像素对应关系图。

表 2-15 LCD 显示缓冲区中的地址与数据表

起始地址	数 据	起始地址	数 据
0000H	A[31:0]	0320H	L[31:0]
0004H	B[31:0]	0324H	M[31:0]
0008H	C[31:0]	0328H	N[31:0]
000CH	D[31:0]	032CH	O[31:0]

【问题 3】(2 分)

S3C44BOX 中内置的 LCD 控制器可支持灰度 LCD 和彩色 LCD 显示。在灰度 LCD 上,使用基于(1)算法和(2)方法,可以支持单色、2 级、4 级和 8 级灰度模式的灰度 LCD 显示。在彩色 LCD 上,可以支持 167 772 色(24 位)。

【问题 4】(3 分)

在 320×240 彩色显示屏中采用 4 位单扫描数据显示模式,图 2-7 中外部接口信号 VFRAME 的频率值为 75 Hz, LCD 显示模式与 MV 模式值的对照见表 2-16。试估算出图 2-7 中 VCLK 信号的频率值。请列出简要的计算式子。

表 2-16 LCD 显示模式与 MV 模式值对照表

灰度等级	显示模式	MV 模式值	灰度等级	显示模式	MV 模式值
单色	4 位单扫描	1/4	16 级	4 位单扫描	1/4
	8 位单扫描	1/8		8 位单扫描	1/8
4 级	4 位单扫描	1/4	彩色	4 位单扫描	1/4
	8 位单扫描	1/8		8 位单扫描	3/8

【问题 5】(3 分)

表 2-17 和表 2-18 分别列出了 PC 和 PD 端口的引脚各功能定义;表 2-19 列出了设置 PC 和 PD 端口各引脚工作模式定义。端口上拉寄存器(PUPC~G)用来设定 PC~PG 这几组端口是否具有内部上拉功能。当 PUP_n 的对应位为 0 时,该引脚上的上拉功能将使能;当为 1 时,该引脚上的上拉功能将不起作用。

表 2-17 PC 端口各引脚功能定义表

端口 PC	可选择的引脚功能		
	功能 1	功能 2	功能 3
PC0	输入/输出	DATA16	IISLRCK
PC1	输入/输出	DATA17	IISDO
PC2	输入/输出	DATA18	IISDI
PC3	输入/输出	DATA19	IISCLK
PC4	输入/输出	DATA20	VD7
PC5	输入/输出	DATA21	VD6
PC6	输入/输出	DATA22	VD5
PC7	输入/输出	DATA23	VD4
PC8	输入/输出	DATA24	nXDACK1
PC9	输入/输出	DATA25	nXDREQ1
PC10	输入/输出	DATA26	nRTS1
PC11	输入/输出	DATA27	nCTS1
PC12	输入/输出	DATA28	TxD1
PC13	输入/输出	DATA29	RxD1
PC14	输入/输出	DATA30	nRTS0
PC15	输入/输出	DATA31	nCTS

表 2-18 PD 端口各引脚功能定义表

端口 PD	可选择的引脚功能	
	功能 1	功能 2
PD0	输入/输出	VD0
PD1	输入/输出	VD1
PD2	输入/输出	VD2
PD3	输入/输出	VD3
PD4	输入/输出	VCLK
PD5	输入/输出	VLINE
PD6	输入/输出	VM
PD7	输入/输出	VFRAME

表 2-19 PC、PD 端口各引脚工作模式定义表

寄存器名称	位数/引脚	端口地址	描 述	复位值
PCONC[31: 0]	2	0x01D20010	00=输入 01=输出 10=功能 2 11=功能 3	0xAAAAAAAA
PCOND[15: 0]	2	0x01D2001C	00=输入 01=输出 10=功能 2 11=功能 3	0x00

请结合 S3C44BOX 芯片外接 LCD 模块的驱动电路原理图(图 2-8)，将以下(3)~(5)空缺处的内容填写完整以实现对相应 I/O 端口控制寄存器的配置。

```
rPCONC = (3)
rPUPC = 0x00;
(4);
rPUPD=(5);
```

二、要点解析

【问题 1】(3 分)

这是一道要求读者掌握嵌入式系统 LCD 显示方式的基本常识题。通常 LCD 显示方式有反射型 LCD、透射型 LCD 和透反射型 LCD 三种，它们之间的区别见表 2-20。

表 2-20 LCD 显示方式说明表

显示方式	说 明
反射型 LCD	底偏光片后面加了一块反射板，它一般在户外和光线良好的办公室使用
透射型 LCD	底偏光片是透射偏光片，它需要连续使用背光源，一般在光线差的环境使用
透反射型 LCD	处于以上两者之间，底偏光片能部分反光，一般也带背光源，光线好时，可关掉背光源；光线差时，可点亮背光源

【问题 2】(4 分)

这是一道要求读者掌握嵌入式系统 LCD 像素与显示缓冲区数据对应关系的理解题。本题的解答思路如下：

- ① 试题中关键信息“4 位双扫描数据显示模式”，是指采用 2 路 4 位并行数据线进行“行”数据连续移位输出，直到整个帧的数据都被移出为止。同理可知，8 位单扫描显示则是采用 8 位并行数据线进行“行”数据连续移位输出，直到整个帧的数据都被移出为止。
- ② 在驱动 4 位双扫描的 LCD 时，VD[3:0]为上部显示区数据，VD[7:4]为下部显示区数据。
- ③ 160×240 单色显示屏的行像素个数有 160 个，列像素个数有 240 个。
- ④ 在视频显示缓冲区中，0000H~0003H 这 4 个字节的存储单元存放着 A[31:0]的数据，该 32 比特的数据对应于 LCD 显示屏的第一行从左上角开始的 32 个(第 1 个至第 32 个)像素的显示状态。依此可推出，B[31:0]的数据对应于 LCD 显示屏的第一行第 33 个至第 64 个像素的显示状态；C[31:0]的数据对应于 LCD 显示屏的第一行第 65 个至第 96 个像素的显示状态；D[31:0]的数据对应于 LCD 显示屏的第一行第 97 个至第 128 个像素的

显示状态。

⑤ 表 2-15 中,十六进制起始地址 0320H 对应的十进制数是 800,即

$$320H = (1100100000)_2 = 2^9 + 2^8 + 2^5 = (512 + 256 + 32)_{10} = (800)_{10}$$

由于 $800/160=5$, 因此 L[31:0] 的数据将对应于 LCD 显示屏的第 5 行第 1 个至第 32 个像素的显示状态; 而 M[31:0]、N[31:0]、O[31:0] 的数据则分别对应于 LCD 显示屏的第 5 行第 33~64 个、第 65~96 个、第 97~128 个像素的显示状态。

⑥ 综合以上分析过程的数据可得, 在 160×240 的单色显示屏中采用 4 位双扫描数据显示模式, 表 2-15 视频显示缓冲区的地址、数据与 LCD 像素对应关系如图 2-9 所示。

LCD屏	
第1行	A[31] A[30] ... A[0] B[31] B[30] ... B[0] C[31] C[30] ... C[0] D[31] D[30] ... D[0] ...
第5行	L[31] L[30] ... L[0] M[31] M[30] ... M[0] N[31] N[30] ... N[0] O[31] O[30] ... O[0] ...

图 2-9 视频显示缓冲区的地址、数据与 LCD 像素对应关系示意图

【问题 3】(2 分)

这是一道要求读者掌握灰度 LCD 显示基本常识的填空理解题。本题的解答思路如下:

① 市面上出售的 LCD 有两种类型: 一种是带有驱动电路的 LCD 显示模块, 这种 LCD 可以方便地与各种低档单片机进行接口; 另一种 LCD 显示屏是没有驱动电路的, 需要与驱动电路配合使用。

② S3C44BOX 中内置的 LCD 控制器可支持灰度 LCD 和彩色 LCD 显示。在灰度 LCD 上, 使用基于时间的抖动算法 TBDA (Time-Based Dithering Algorithm) 和帧速率控制 FRC (Frame Rate Control) 方法, 可以支持单色、2 级、4 级和 8 级灰度模式的灰度 LCD 显示。在彩色 LCD 上, 同时用到了颜色查找表可以支持 167 772 色 (24 位) 显示。

③ FRC 方法实现灰度级在 STN 屏幕上的显示。STN 屏显示灰度级的机理就是 FRC。例如, 为了显示 16 级灰度中的第 3 级 (3/16), 3 个时间单位用于显示, 13 个时间单位不显示。换句话说, 必须从 16 帧中选择 3 帧出来, 这 3 帧中相应的像素显示, 而其余的 13 帧中的相应像素不显示, 而且 16 帧必须是周期显示的。这就是在屏幕上显示灰度的基本原理, 即帧速率控制灰度显示的原理。

④ 在 STN 显示时, 应注意到噪声也会产生积累。例如, 如果所有的像素在第一帧全部显示而在下一帧又全部关闭, 噪声最大化。为了减少噪声, 应尽可能地使用概率均等的原则。具体实现中, 使用了基于时间的抖动算法 (TBDA)。对于 16 级灰度, 在灰度级和 FRC 之间, FRC 应该有下面的关系: 第 15 级灰度应该一直显示; 第 14 级显示 6 次, 关闭 1 次; 第 13 级显示 4 次, 关闭 1 次; 第 0 级一直关闭。

【问题 4】(3 分)

这是一道要求读者掌握根据 LCD 尺寸、显示模式等外界参数确定 VCLK 信号频率的计算题。本题的解答思路如下:

① VCLK 信号的频率可以通过 LCDCON1 寄存器的 CLKVAL 域来确定。它们存在以下关系:

$$VCLK(Hz) = MCLK / (CLKVAL \times 2)$$

式中, MCLK 是指系统时钟输出的频率值。由上述关系知, CLKVAL 的值决定了 VCLK 的频率, 为了确定 CLKVAL 的值, 应该计算一下 LCD 控制器向 VD 端口传输数据的速率, 使得 VCLK 的值大于数据传输的速率。

② 数据传输速率 R 的计算公式是

$$R = HS \times VS \times FR \times MV$$

式中, HS 是 LCD 的行的像素点数值; VS 是 LCD 的列的像素点数值; FR 是帧速率; MV 是模式值。

③ 由题干的关键信息“在 320×240 彩色显示屏中”可知, 该 LCD 有 320 个行的像素点值, 有 240 个列的像素点值。

④ 根据题干的关键信息“彩色显示屏中采用 4 位单扫描数据显示模式”并查表 2-16 得, 该 LCD 的 MV 模式值为 $3/4$ 。

⑤ 由此可得 LCD 控制器向 VD 端口传输数据的数据传输速率为

$$R = HS \times VS \times FR \times MV = 320 \times 240 \times 75 \times 0.75 = 4\,320\,000 \text{ Hz} = 4.32 \text{ MHz}$$

⑥ 为了使 LCD 控制器支持已有的 LCD 驱动器, 通常 LCD 控制器的最大 VCLK 频率可达 16.5 MHz。在具体工程项目设计时, VCLK 的值应大于 2 M, 小于 16 M, 因此 CLKVAL 可以取值 3~15。

⑦ 另外补充一点, 在图 2-7 中, VFRAME 和 VLINE 脉冲的产生是通过对 LCD-CON2 寄存器的 HOZVAL 和 LINEVAL 域进行配置来完成的。每个域都与 LCD 的尺寸和显示模式有关。VFRAME 信号的频率还与 LCDCON1 和 LCDCON2 寄存器中的 WLH (VLINE 脉冲宽度)、WHLY (VLINE 脉冲之后 VCLK 的延时宽度)、HOZVAL、VLINE-BLANK 和 LINEVAL 有关。

【问题 5】(3 分)

这是一道要求读者掌握嵌入式系统微处理器 I/O 端口初始化的分析题。本题的解答思路如下:

① 根据图 2-8 中 S3C44BOX 芯片与 LCD 模块的驱动电路相关的引脚定义, 查阅表 2-17 可知 PC 端口的 PC4~PC7 引脚工作于功能 3 模式, 查阅表 2-18 可知 PD 端口的 PD0~PD7 引脚工作于功能 2 模式。

② 由于 PC 端口的 PC4~PC7 引脚工作于功能 3 模式, 因此查阅表 2-19 可知与 PC4~PC7 各引脚的相关寄存器位应设置成“11”状态。同理, 由 PD 端口的 PD0~PD7 引脚工作于功能 2 模式, 查阅表 2-19 可知与 PD4~PD7 各引脚的相关寄存器位应设置成“10”状态。

③ 根据以上分析结果, 将表 2-17 和表 2-19 与本题 PC 端口相关的关键信息整理成表 2-21。将表 2-18 和表 2-19 与本题 PD 端口相关的关键信息整理成表 2-22。

表 2-21 PC 端口各引脚功能及其工作模式定义表

端口 PC	功能 3	相应的寄存器状态位	位状态值
PC4	VD7	PCONC[9: 8]	11
PC5	VD6	PCONC[11: 10]	11
PC6	VD5	PCONC[13: 12]	11
PC7	VD4	PCONC[15: 14]	11

表 2-22 PD 端口各引脚功能及其工作模式定义表

端口 PD	功能 2	相应的寄存器状态位	位状态值
PD0	VD0	PCOND[1: 0]	10
PD1	VD1	PCOND[3: 2]	10
PD2	VD2	PCOND[5: 4]	10
PD3	VD3	PCOND[7: 6]	10
PD4	VCLK	PCOND[9: 8]	10
PD5	VLINE	PCOND[11: 10]	10
PD6	VM	PCOND[13: 12]	10
PD7	VFRAME	PCOND[15: 14]	10

④ 查阅表 2-21 得知, PC 端口使用 1 个 32 位端口配置寄存器(PCONC)定义各引脚的功能, 且每个引脚均使用 2 位寄存器位来定义各个引脚功能的配置方式。要将 PC4~PC7 各引脚定义为第 3 功能状态, 需将端口配置寄存器(PCONC)的第 8~15 位置 1, 因此(3)空缺处可填入二进制数“00 00 00 00 00 00 00 00 11 11 11 11 00 00 00 00”。由于本试题中对 PCONC 其他各引脚的功能没有特别的说明, 因此可将 PCONC 中该部分的寄存器位做清零处理, 另外(3)空缺处也可填入等价的十六进制数“0x0000FF00”。

⑤ 由试题中关键信息“端口上拉寄存器(PUPC~G)用来设定 PC-PG 这几组端口是否具有内部上拉功能”以及图 2-8 电路原理图可知, 端口上拉寄存器 PUPC、PUPD 分别用来设定 PC、PD 端口是否具有内部上拉功能。

⑥ 试题中已给出“PC=0x00;”语句, 其作用是使能 PC 端口内部的上拉功能。同理可知, 如要使能 PD 端口内部的上拉功能, 则需将端口上拉寄存器 PUPD 的所有位清零。因为电路原理图(图 2-8)中使用了 PD 端口的所有引脚。由此确定(5)空缺处可填入“0x00”。

⑦ 由于题目要求实现对图 2-8 电路图中相应 I/O 口控制寄存器的配置, 而图 2-8 中使用到 PC 和 PD 端口, 因此可确定(4)空缺处所填写的内容与端口配置寄存器(PCOND)相关。

查阅表 2-22 得知, PD 端口使用 1 个 16 位端口配置寄存器(PCOND)定义各引脚的功能, 且每个引脚均使用 2 位寄存器位来定义各个引脚功能的配置方式。要将 PD0~PD7 各引脚定义为第 2 功能状态, 需将端口配置寄存器(PCOND)的状态值从高至低依次设置为“10 10 10 10 10 10 10 10”, 因此(4)空缺处可填入“rPCOND=0x0AAAA;”语句, 其中二进制数“10 10 10 10 10 10 10 10B”也可以等价的十六进制数“0x0AAAA”填入。

⑧ 以下给出 PC 及 PD 端口完整的 I/O 口初始化程序及其解释。

```

rPCONC=0x0000FF00;    //设置 PC 口工作在第 3 功能状态
rPUPC=0x00;            //使能 PC 端口内部的上拉功能
rPCOND=0x0AAAA;        //设置 PD 口工作在第 2 功能状态
rPUPD=0x00;            //使能 PD 端口内部的上拉功能

```

三、参考答案

表 2-23 给出了本案例问题 1 至问题 5 的参考答案,供读者练习时参考,以便查缺补漏。读者也可依照所给出的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

表 2-23 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (3 分)	反射型 LCD、透射型 LCD、透反射型 LCD (每点 1 分)	
【问题 2】 (4 分)	参见图 2-9 (完整画出其中一行得 2 分,注意各数据位的先后顺序,未标出“第 5 行”扣 1 分)	
【问题 3】 (2 分)	(1) 基于时间的抖动算法(或 TBDA 算法) (2) 帧速率控制方法(或 FRC 方法) (每空 1 分)	
【问题 4】 (3 分)	VCLK 值要大于 LCD 控制器向 VD 端口传输数据的数据传输速率 R(1 分), $R=HS\times VS\times FR\times MV=320\times 240\times 75\times 0.75=4\,320\,000\text{ Hz}=4.32\text{ MHz}$ (2 分)	
【问题 5】 (3 分)	(3) 0x0000FF00, 或等价的二进制数形式 (4) rPCOND=0x0AAAA, 其中“0x0AAAA”也可是等价的二进制数形式 (5) 0x00 (每空 1 分)	

2.2.4 案例 4 串口通信技术

一、案例描述

串行通信是微计算机之间一种常见的近距离通信手段,因使用方便且编程简单而广泛使用,几乎所有的嵌入式微控制器和 PC 都提供串行通信接口。阅读以下关于嵌入式系统串口通信技术的说明,结合电路原理图回答问题 1 至问题 5。(15 分)

【说明】

电子工业协会(EIA)推荐的串行数据传输总线标准——RS-232-C 标准,采用的接口是 9 芯或 25 芯的 D 型插头,以常用的 DB-9 插头为例,各引脚定义如表 2-24 所示。

表 2-24 DB-9 插头引脚信号描述表

引 脚	名 称	功 能 描 述	引 脚	名 称	功 能 描 述
1	DCD	数据载波检测	6	DSR	数据设备准备好
2	RXD	数据接收	7	RTS	请求发送
3	TXD	数据发送	8	CTS	清除发送
4	DTR	数据终端准备好	9	RI	振铃指示
5	GND	地			

RS-232-C 标准所定义的高、低电平信号与 S3C4510B 系统的 LVTTL 电路所定义的高、低电平信号存在较大的差异,即 LVTTL 的标准逻辑“1”对应 2~3.3 V 电平,标准逻辑“0”对应 0~0.4 V 电平;而 RS-232-C 标准采用负逻辑方式,标准逻辑“1”对应 -3~-25 V 电平,标准逻辑“0”对应 +3~+25 V 电平。显然,两者间要进行通信必须经过信号电平的转换,目前常使用的电平转换芯片是 MAX232。使用 MAX3232 芯片实现的串口电平转换的电路原理图如图 2-10 所示。

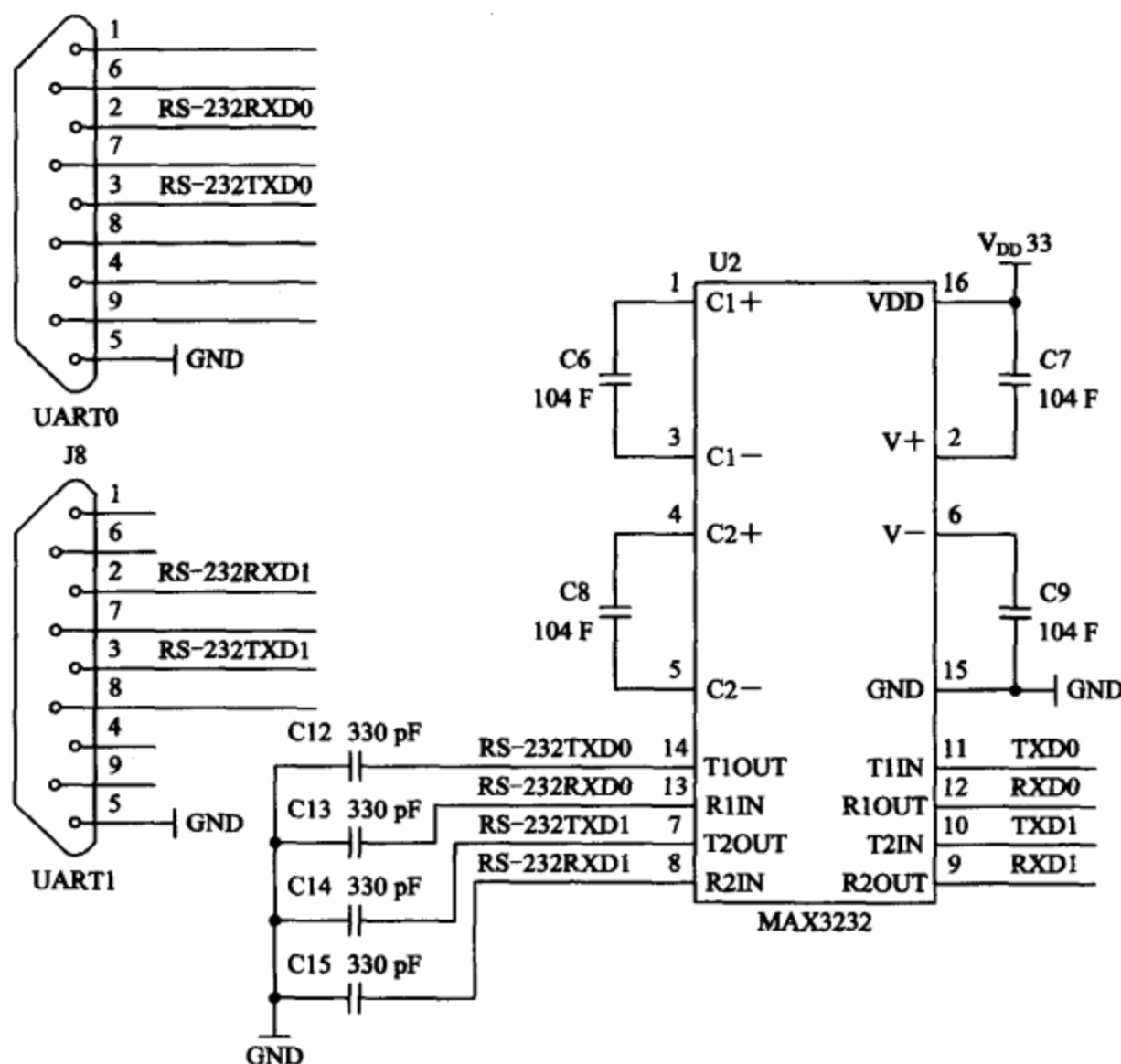


图 2-10 使用 MAX3232 芯片实现的串口电平转换的电路原理图

S3C44BOX 芯片的通用异步收发器(UART 单元)提供两个独立的异步串行 I/O 端口,每个通信端口均可工作在中断模式或 DMA 模式, UART 能产生内部中断请求或 DMA 请求在 CPU 和串行 I/O 口之间传送数据。它们支持的最高波特率为 115.2 kb/s。每个 UART 通道包含两个 16 位 FIFO 缓冲区分别提供给接收和发送。每一个异步串行通信端口都具有独立的波特率发生器、发送器、接收器和控制单元。波特率发生器可由片内系统时钟 MCLK 驱动,或由外部时钟 UCLK(Pin64)驱动。发送器和接收器都有独立的数据缓冲寄存器和数据移位器。待发送的数据首先传送到发送缓冲寄存器,然后拷贝到发送移位器并通过发送数据引脚 UATXD_n 发送出去。接收数据首先从接收数据引脚 UARXD_n 移入移位器,当接收到一个字节时就拷贝到接收缓冲寄存器。

S3C44BOX 的 UART 可以进行以下参数的设置,如可编程的波特率,红外收/发模式,1 或 2 个停止位,5 位、6 位、7 位或 8 位数据宽度和奇偶位校验等。

【问题 1】(2 分)

串行通信就是使数据一位一位地进行传输而实现的通信。当采用异步通信时,通信的字符格式必须遵循哪些规定?

【问题 2】(3 分)

在近距离通信系统中,无需使用电话线和 Modem,可以直接进行端到端的连接,即使用 Null Modem 连接电缆。请结合表 2-24 中 DB-9 插头各引脚信号描述,按 Null Modem 电缆连接方式将图 2-11 中空缺的接线补充完整。

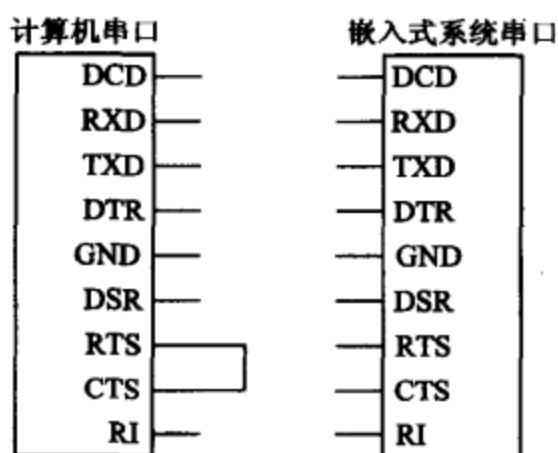


图 2-11 Null Modem 通信连线图

【问题 3】(2 分)

以 960 字符/秒的异步通信速率传送 ASCII 码,一帧数据带有 7 位 ASCII 编码、1 位奇校验位和 1 位停止位,求该嵌入式系统的数据传送的波特率。

【问题 4】(3 分)

S3C44BOX 的每个 UART 模块包含有波特率发生器、发送器、接收器和控制单元等部件。其中波特率发生器以 MCLK 作为时钟源,每个 UART 的波特率发生器为传输提供了串行移位时钟。波特率时钟由通过时钟源的 16 分频及一个由 UART 波特率除数寄存器 (UBRDIV_n)指定的 16 位除数决定。UBRDIV_n 的值可以按照下式确定:

$$\text{UBRDIV}_n = (\text{round_off}) \left(\frac{\text{MCLK}}{\text{bps} \times 16} \right) - 1$$

求该嵌入式系统采用串行数据传输最高波特率为 115 200 b/s 时,系统主频至少为多少赫兹?

【问题 5】(5 分)

UART 线控制寄存器主要用来规定传输帧的格式,线控制寄存器各位定义如表 2-25 所示。UCON_n 控制寄存器各位定义如表 2-26 所示。波特率除数寄存器 UBRDIV_n 各位定义如表 2-27 所示。

表 2-25 ULCONn 线控制寄存器

位	位 名	功 能 描 述	初始值
[1:0]	每帧长度	此两位数值指示发送或接收的每帧的数据位: 00=5 位, 01=6 位, 10=7 位, 11=8 位	00
[2]	停止位数	该位指示每帧数据的停止位数: 0= 每帧 1 个停止位, 1= 每帧 2 个停止位	0
[5:3]	奇偶校验模式	此三位数值指示在数据的发送与接收过程中如何生成校验并进行检测: 0xx=无校验, 100=奇校验, 101=偶校验, 110=强制置 1, 111=强制清零	000
[6]	红外模式选择	是否采用红外通信模式, 0=正常模式操作, 1=红外接收/发送模式	0
[7]	保留		0

表 2-26 UCONn 控制寄存器

位	位 名	功 能 描 述	初始值
[1:0]	接收模式选择	该两位的值决定当从 UART 接收缓冲寄存器中读取数据时的当前功能: 00=禁止 Rx 模式, 01=产生中断请求, 10=产生 BDMA 0 请求(仅 UART0), 11=产生 BDMA 1 请求(仅 UART1)	00
[3:2]	发送模式选择	该两位的值决定当写数据到 UART 发送缓冲寄存器中的时的当前功能: 00=禁止 Tx 模式, 01=产生中断请求, 10=产生 BDMA 0 请求(仅 UART0), 11=产生 BDMA 1 请求(仅 UART1)	00
[4]	发送终止信号	设置该位, 令 UART 在一帧时间中发送一个终止状态。发送完毕系统自动清除该位。0=正常传输, 1=发送终止信号	0
[5]	回环模式选择	该位选择 UART 是否进入回环模式。在回环模式下, 发送的数据的输出置为高电平, 发送缓冲寄存器 UTXBUF 在内部直接连接到接收缓冲寄存器 URXBUF。回环模式仅用于测试目的, 在正常操作模式下, 该位应为 0	0
[6]	接收错误状态中断使能	使能 UART 在接收操作中发生错误时的错误中断响应: 0=不产生错误状态中断请求, 1=产生错误状态中断请求	0
[7]	接收超时中断使能	在 UART 的 FIFO 使能的情况下, 使能/禁止接收超时中断: 0=禁止, 1=允许	0
[8]	接收中断类型	中断请求类型: 0=脉冲(在接收缓冲区接收到数据时立即引发中断), 1=电平(在接收缓冲区正在接收到数据时引发中断)	0
[9]	发送中断类型	中断请求类型: 0=脉冲(在发送缓冲区变空时立即引发中断), 1=电平(在发送缓冲区变空时引发中断)	0

表 2-27 UBRDIVn 寄存器

位	位 名	功 能 描 述	初始值
[15:0]	UART 波特率除数	波特率除数的值=(int)(MCLK/16/Baudrate)-1)	—

图 2-12 给出了一段嵌入式系统通过串行口发送数据的 C 语言程序代码，该程序的通信协议为：19 200 波特、8 位数据、1 位停止位及无校验位。请将(1)~(3)空缺处的注释填写完整，并说明图 2-12 所示的 C 语言程序完成的功能。

```
void InitUART(int Port,int Baudrate);
void PrintUART(int Port,char * s);
int Main()
{
    InitUART(0,0x4B00);          //(1)
    for(;;)D    {
        PrintUART(0,"Communcation Testting! \r\n");
    }
    return(0);
}

void PrintUART(int Port,char * s)
{
    if(Port==0)
        for(; * s! ='\0';s++)
        {
            for(,(! (USTAT0&0x40)););
            UTXBUF0= * s;
        }
    if(Port==1)D    for(; * s! ='\0';s++)
    {
        for(,(! (USTAT1&0x40)););
        UTXBUF1= * s;
    }
}

void InitUART(int Port,int Baudrate)
{
    if(Port==0)          //串行端口选择
    {
        ULCON0=0x03;      //(2)
        UCON0=0x245;      //(3)
        UBRDIV0=((int)(MCLK/16/Baudrate)-1);    //根据波特率计算 UBRDIV0 的值
    }
    if(Port==1)
```



```

{
    ULCON1=0x03;D          UCON1=0x245;
    UBRDIV1=((int)(MCLK/16/Baudrate)-1);    //根据波特率计算 UBRDIV0 的值
}
}

```

图 2-12 某串口通信程序代码

二、要点解析

【问题 1】(2 分)

这是一道要求读者掌握串行通信基本常识的简答题。本题所涉及的知识点如下：

① 串行通信主要有三种数据传送模式，其中单工通信是指通信线路上的数据只能按单一方向传送；半双工通信是指通信线路上的数据允许从一方传送给另一方，也允许数据反向传送，但不能在同一时刻进行双向通信；全双工通信是指一个通信线路上允许数据同时双向通信。

② 异步通信方式是指以一个字符为传输单位，通信中两个字符间的时间间隔是不固定的，然而在同一个字符中的两个相邻位代码间的时间间隔是固定的。

③ 异步通信时必须遵循的三项规定分别是字符的格式、波特率和校验位。其中，字符格式规定每个字符传送时，必须前面加 1 位起始位，后面加 1 位(或 1.5 位，或 2 位)停止位。

④ 例如，以异步通信方式传送 ASCII 码时，一帧数据共有 10 位，分别由 1 个起始位、7 位 ASCII 编码、1 个奇偶校验位和 1 个停止位组成。

【问题 2】(3 分)

这是一道要求读者掌握串行通信零调制解调器电缆接线的实践操作题。本题的解答思路如下：

① 由于这两种设备必须分别以 DTE 和 DCE 方式成对出现才符合 RS-232C 标准接口的要求，因此，当需要用 RS-232C 接口直接连接两台近距离设备时，就需要借助于一种采用交叉跳接信号线方法的连接电缆，使得连接在电缆两端的 DTE 通过电缆看对方都好像是 DCE 一样，从而满足 RS-232C 接口需要 DTE-DCE 成对使用的要求。这根连接电缆称做零调制解调器，又称空 Modem 或虚拟 Modem(Null Modem)。

② 在图 2-11 中最精简的 Null Modem 接线方法是将计算机串口第 2 引脚(RXD 接收信号线)连接至嵌入式系统串口的第 3 引脚(TXD 发送信号线)，将计算机串口第 3 引脚(TXD 信号线)连接至嵌入式系统串口的第 2 引脚(RXD 信号线)，然后将两个串口第 5 引脚(GND 信号地线)直接连接。

③ 同时注意到，在图 2-11 中计算机串口的第 7 引脚(请求发送 RTS 信号线)与本串口的第 8 引脚(清除发送 CTS 信号线)直连，由此可知试题要求画出的是完整的 Null Modem 接线图。

④ 完整的 Null Modem 接线方法还需将计算机串口的第 4 引脚(数据终端准备好 DTR 信号线)与本串口的第 6 引脚(数据端接设备准备好 CTS 信号线)直连，将嵌入式系统串口的第 4 引脚(DTR 信号线)与其自身的第 6 引脚(CTS 信号线)直连，将嵌入式系统串口的

第7引脚(RTS信号线)与其自身的第8引脚(CTS信号线)直连。

⑤ 完整的零调制解调器通信连线图如图2-13所示。

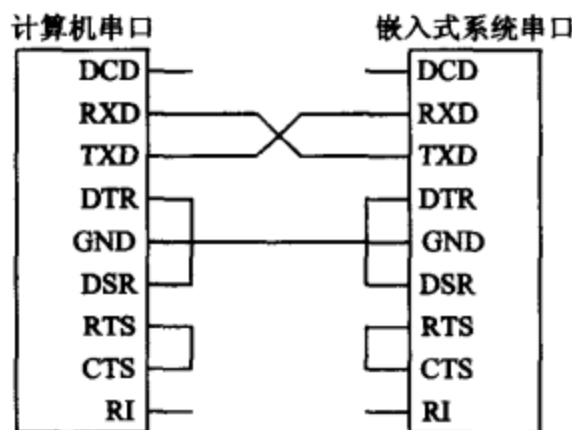


图 2-13 完整的 Null Modem 通信连线图

【问题 3】(2 分)

这是一道要求读者掌握异步通信方式波特率计算方法的应用题。本题的解答思路如下：

① 由于异步通信方式的字符格式规定，每个字符传送时，必须在前面加 1 位起始位。因此传送一帧 ASCII 码所包含的数据位共有 10 位(1+7+1+1=10)。

② 由于异步通信方式是指以 1 个字符为传输单位，而本试题中 1 个字符包含有 10 位数据，因此该嵌入式系统的数据传送的波特率=960 字符/秒×10 位=9600 b/s=9600 波特。

【问题 4】(3 分)

这是一道要求读者理解 UART 波特率除数寄存器数值计算的综合应用题。本题的解题思路如下：

① 在计算式子

$$UBRDIV_n = (\text{round_off}) \left(\frac{MCLK}{\text{bps} \times 16} \right) - 1$$

中，(round_off)的作用是对其后圆括号中计算式子做取整操作；而 $\frac{MCLK}{\text{bps} \times 16}$ 式子的除数的范围为 $1 \sim (2^{16} - 1)$ 。

② 当 UBRDIV_n 取值不变时，系统主频(MCLK)与波特率(bps)成反比关系。

③ 当 UBRDIV_n = 0 时，MCLK = bps × 16。将试题中串行数据传输最高波特率 115 200 b/s 代入可得

$$MCLK = \text{bps} \times 16 = 115\,200 \times 16 = 1.8432 \text{ MHz}$$

【问题 5】(5 分)

这是一道要求读者掌握对 UART 口进行初始化设置的综合理解题。本题的分析思路如下：

① (1)空缺处是对“InitUART(0, 0x4B00);”语句的解释，而 InitUART 函数已在 main 函数前事先声明。根据自定义函数“void InitUART(int Port, int Baudrate)”传送的参数，以及试题中关键信息“通信协议为：19 200 波特、8 位数据位、1 位停止位、无校验位”可推理出，“InitUART(0, 0x4B00);”语句是对串行口 1 进行初始化工作，且将串口通信的波特率设置为 19 200 Baud，因为与十六进制数 0x4B00 等价的十进制数是 19 200。

② 对于“ULCON0=0x03;”语句中十六进制数 0x03 等价的二进制数是0000 0011。查表 2-25 可知, 串行口 1 通信时不采用红外线传输模式, 无奇偶校验位, 有 1 个停止位, 每帧携带 8 个数据位。

③ 同理, 对于“UCON0=0x245;”语句, 与十六进制数 0x245 等价的二进制数是 0010 0100 0101。查表 2-26 可知, 串行口 1 通信时发送中断为电平触发方式(在发送缓冲区变空时引发中断), 接收中断为边沿脉冲触发方式(在接收缓冲区接收到数据时立即引发中断), 禁止产生接收超时中断, 允许产生接收错误状态中断, 禁止回送模式, 禁止终止信号, 发送模式为中断请求模式, 接收模式也为中断请求模式。

④ 由 main 函数中“PrintUART(0, "Communcation Testting! \r\n");”语句可知, 该语句向自定义函数“void PrintUART(int Port, char * s)”传送的通信端口参数是 0, 传送的字符串是“Communcation Testting!”, 而该语句是“for(;;)”无限循环体的执行语句。因此, 如将图 1-22 的 C 语言代码添加到新建的项目, 并对该项目进行编译链接, 则生成可执行的映像文件。当可执行的映像文件运行时, 嵌入式系统将会不停地通过 UART0 对外发送字符串“Communcation Testting!”。

三、参考答案

表 2-28 给出了本案例问题 1 至问题 5 的参考答案, 供读者练习时参考, 以便查缺补漏。读者也可依照所给出的评分标准得出自己的测试分数, 从而大致评估自己对这一方面知识点的掌握程度。

表 2-28 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (2 分)	每个字符传送时, 必须前面加 1 位起始位(1 分), 最后加入 1 位(或 1.5 位, 或 2 位)停止位(1 分)	
【问题 2】 (3 分)	参见本问题要点解析中的图 2-13 (每条连线 0.5 分)	
【问题 3】 (2 分)	传送一帧 ASCII 码所包含的数据位共有 10 位(1 分) 波特率=960 字符/秒×10 位=9600 b/s=9600 波特(1 分)	
【问题 4】 (3 分)	$MCLK = bps \times 16 = 115\ 200 \times 16 = 1.8432\ \text{MHz}$ (3 分)	
【问题 5】 (5 分)	(1) 对串行口 1 进行初始化工作, 且将串口通信的波特率设置为 19 200 Baud(1 分) (2) 设置串行口 1 通信时不采用红外线传输模式, 无奇偶校验位, 1 个停止位, 每帧携带 8 个数据位(1 分) (3) 串行口 1 通信时发送中断为电平触发方式, 接收中断为边沿脉冲触发方式, 允许产生接收错误状态中断, 禁止产生接收超时中断, 禁止回送模式, 禁止终止信号, 发送模式、接收模式均为中断请求模式(1 分) 该程序功能: 通过 UART0 不停地对外发送字符串“Communcation Testting!”(2 分)	

2.2.5 案例5 网络接口技术

一、案例描述

嵌入式 Internet 是近几年随着计算机网络技术的普及而发展起来的一项新兴概念和技术,它通过为现有嵌入式系统增加因特网接入功能来扩展其信息交互的能力。随着嵌入式 Internet 产品的开发和应用,远程监控、数据采集和系统重构等领域将会发生深刻的变革,将推动嵌入式系统的更多潜在的应用。

阅读以下关于嵌入式系统网络接口技术的说明,结合电路原理图回答问题 1 至问题 5。(15 分)

【说明】

Embedded Network(嵌入式网络)能够实现生产设备内部多个分布式子系统之间的实时通信,已成为现代化工业企业网络环境的一个组成要素。为适应嵌入式分布处理结构和应用上网需求,嵌入式系统必须配有一种或多种网络通信接口,使嵌入式微控制器工作不仅执行传统的控制功能,而且还执行连接因特网相关的功能,从而把标准网络技术一直扩展到嵌入设备,由嵌入式系统自身实现 Web 服务器功能。

RTL8019 是一种高度集成的以太网控制器,共有 32 个输入/输出地址,对应地址偏移量为 0x00~0x1F。RTL8019 的内部寄存器是分页的,每个寄存器都是 8 位。以太网控制器含有 16 KB 的 RAM,地址为 0x4000~0x7FFF,每 256 个字节称为一页,共有 64 页。使用 0x40~0x45 作为以太网控制器的发送缓冲区,共 6 页。使用 0x46~0x5F 作为以太网控制器的接收缓冲区,共 32 页。CURR 和 BNRy 寄存器是以太网数据收发过程中用到的两个最主要的寄存器,CURR 和 BNRy 主要用来控制缓冲区的存取过程,保证能顺次写入和读出。CURR 是以太网控制器写接收缓冲区的指针。BNRY 指向接收缓冲区中已经被读取的最后一个页。

与常规的网卡设计思路不同的是,在嵌入式系统中,系统的精简一直是个主要的原则。在嵌入式系统设计时,配置 RTL8019 为非即插即用模式,使用固定的中断及固定的端口地址,从而可以节省配置存储器,减小嵌入式硬件平台的体积。一种完成以上设计思想的嵌入式微处理器 S3C44BOX 与 RTL8019 以太网模块的接口电路如图 2-14 所示。

【问题 1】(4 分)

嵌入式 Internet 技术涉及的两个关键问题是传送信息的媒体和所采用的通信协议。目前最常用的传输媒体是非屏蔽双绞线(UTP),广泛采用的通信协议是 TCP/IP 协议。其网络体系结构与协议分层如图 2-15 所示。请将图 2-15 中(1)、(2)空缺处的内容填写完整,并结合图 2-14 的电路原理图说明 S3C44BOX 微处理器和 RTL8019 芯片各完成 TCP/IP 协议簇中哪些层次的功能?

【问题 2】(2 分)

在 TCP/IP 协议簇的传输层主要有传输控制协议 TCP 和用户数据报协议 UDP 两种。应用于工业控制领域的嵌入式系统在传输层应考虑采用哪种传输层协议?请用 150 字以内的文字简要说明理由。

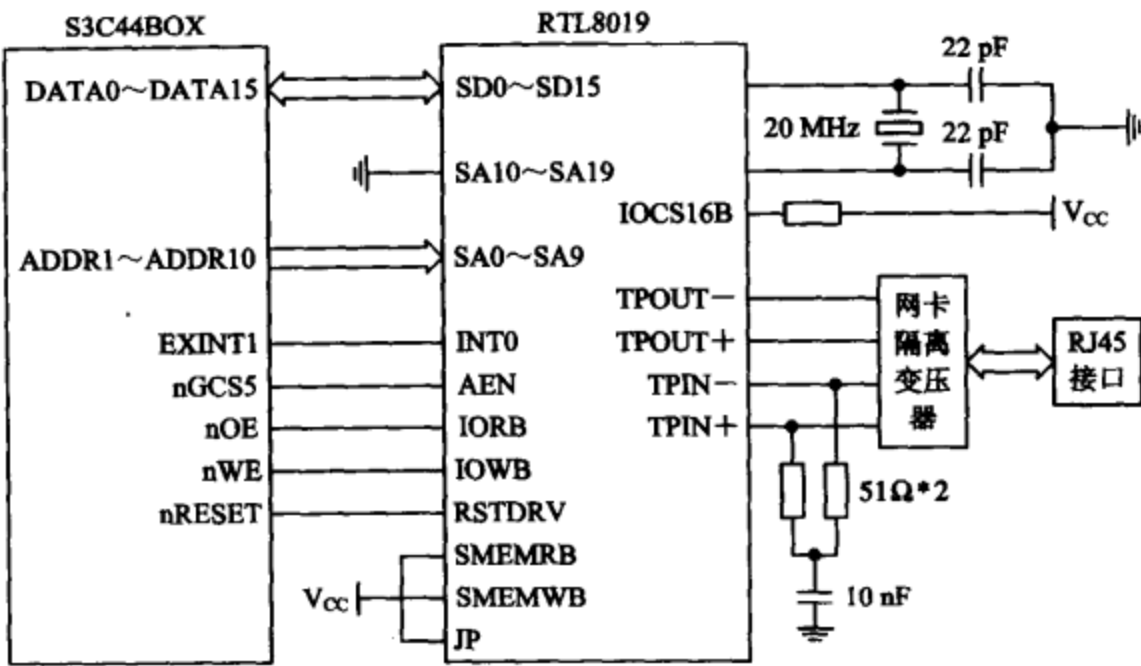


图 2-14 S3C44BOX 与 RTL8019 以太网模块的接口电路原理图

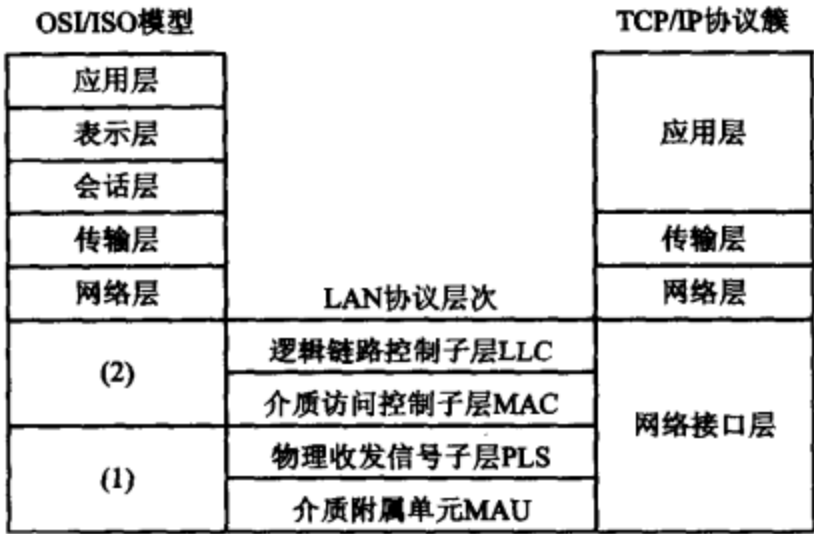


图 2-15 网络体系结构与协议分层示意图

【问题 3】(3 分)

在嵌入式系统中增加以太网接口的实现方法有：① 嵌入式处理器结合网卡芯片；② 直接采用带有以太网接口的嵌入式处理器。请用 200 字以内的文字简要比较这两种实现方法的优缺点。

【问题 4】(3 分)

图 2-16 给出了 IEEE 802.3 标准以太网 MAC 帧的结构，请用方向箭头(→)标出 MAC 帧各字段发送的先后次序。在图 2-14 电路原理图中通过 RTL8019 传输数据时，以太网 MAC 帧的哪些字段是 RTL8019 芯片自动产生的？哪些字段是需要上层软件配置处理的？

56位	8位	48位	48位	16位	46~1500字节	0~46字节	32位
帧前导码	帧首定界符	目的地址	源地址	类型字段	数据段	填充位	帧校验码

图 2-16 IEEE 802.3 标准 MAC 帧结构

【问题 5】(3 分)

图 2-16 电路原理图中示意了网络接口控制芯片 RTL8019AS 的一种复位方法,即通过将其复位引脚(3)进行冷复位;也可通过读写其复位端口 RSTPORT 来进行热复位。另一种复位方法是(4)。

在完成对寄存器的初始化后,还要对以太网控制器 RTL8019AS 的物理地址进行设置。该物理地址在 RAM 地址的 0x0000~0x000B 空间中是按(5)存储的。

二、要点解析**【问题 1】(4 分)**

这是一道要求读者掌握嵌入式系统网络协议层次的基本常识题。本题的解答思路如下:

① 国际标准化组织(ISO)于 1979 年研究制定了一种开放系统互连的体系结构(Open Systems Interconnection),它定义了连接异种计算机的标准框架。OSI 参考模型分为 7 层,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。

② 物理层位于 OSI 参考模型的最底层,其任务是为它的上一层提供一个物理连接,以及它们的机械、电气、功能和过程特性。该层的数据协议单元(PDU)是比特,服务访问点(SAP)是网络接口。该层在局域网协议标准中又被分成了物理收发信号子层(PLS)和介质附属单元(MAU)。可见(1)空缺处应填入物理层。

③ 数据链路层负责在两个相邻节点间的线路上无差错地进行帧的透明传送。和物理层相似,数据链路层要负责建立、维持和释放数据链路的连接。该层的数据协议单元(PDU)是帧,服务访问点(SAP)是物理地址(即 MAC 地址)。该层在局域网协议标准中又被分成了逻辑链路控制子层(LLC)和介质访问控制子层(MAC)。可见(2)空缺处应填入数据链路层。

④ 在 OSI 参考模型中网络层的任务是选择合适的网间路由和交换节点,以确保数据及时传送。网络层的数据协议单元(PDU)是 IP 包,服务访问点(SAP)是 IP 地址。传输层的任务是根据通信子网的特性最佳地利用网络资源,并以可靠和经济的方式,为两个端系统(也就是源站和目的站)的会话层之间,提供建立、维护和取消传输连接的功能,负责可靠地传输数据。传输层的数据协议单元(PDU)是数据段,服务访问点(SAP)是端口号。

⑤ 在 OSI 参考模型中会话层提供包括访问验证、会话同步和会话管理在内的建立和维护应用之间通信的机制。表示层主要解决信息的语法表示、数据的加密和解密、压缩和解压缩等问题。应用层用于确定进程之间通信的性质,以满足用户需要以及提供网络与用户应用软件之间的接口服务。

⑥ 在嵌入式系统中,网络接口控制器(NIC)处于 OSI 参考模型的物理层和数据链路层上(大致对应于 TCP/IP 协议簇的网络接口层),用来实现网络控制逻辑和与处理器的接口逻辑。图 2-14 中 RTL8019AS 以太网控制器在逻辑链路控制子层(LLC)向高层提供逻辑接口,具有帧发送和接收功能。发送时把要发送的数据加上地址和 CRC 校验构成帧;接收时把帧拆开,执行地址识别和 CRC 校验,并具有帧顺序控制、差错控制和流控制等功能。介质访问控制子层(MAC)管理链路上的通信和各节点之间通信如何控制,以及完成 IEEE 802.3 标准采用的带冲突检测的载波监听多路访问 CSMA/CD 来处理信道上的冲突问题。

在物理层,根据 IEEE 802.3 标准规定的信号编码和介质,建立物理连接,包括比特流的传输和接收、同步字符的产生与删除等。

⑦ 在图 2-14 中, S3C44BOX 微处理器中运行嵌入式 TCP/IP 协议通信模块来实现与 Internet 的连接,负责完成 TCP/IP 协议中网络层、传输层和应用层的服务功能。

⑧ 完整的网络体系结构与协议分层示意图如图 2-17 所示。

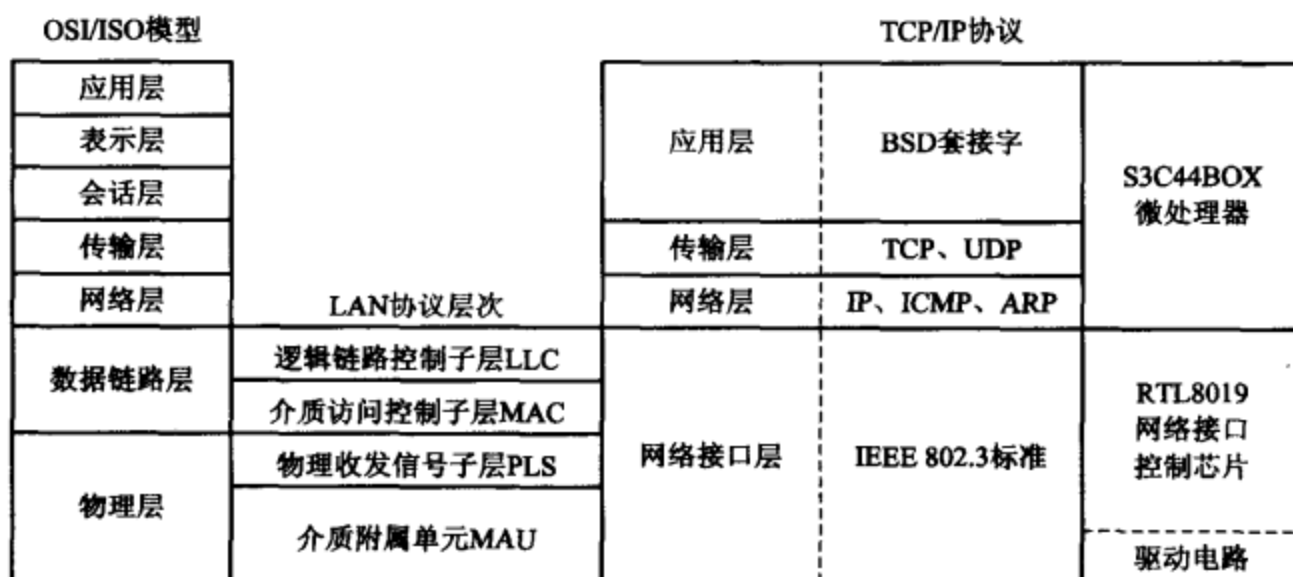


图 2-17 网络体系结构与协议分层示意图

【问题 2】(2 分)

这是一道要求读者掌握 TCP 协议和 UDP 协议应用特点的综合题。本题的分析思路如下:

① 由于 TCP/IP 协议簇中传输层的底层是网络层,而 IP 协议是网络层的基础,它为不同网络的主机之间发送数据报的操作序列,提供的是无连接、不可靠的服务,因此嵌入式系统只把 IP 协议作为传输工具进行简化,以完成主要的操作。

② 在工业控制领域,实时监控系统中的数据传输量并不大,可靠性要求却较高。TCP 协议是一种面向连接、端对端的可靠通信协议,它采用了诸如三次握手建立连接,通过序列号来解决通信时的丢失、重复、失序问题,并有严格的连接建立、复位、关闭步骤等机制来保证传输可靠性,从而弥补了 IP 协议的不足,保证了数据流的可靠传输。

③ UDP 协议是一种无连接、不可靠的传输层通信协议,它只是把应用程序传来的数据加上 UDP 头作为 UDP 数据包发送出去,而不保证它们是否能到达目的地。该协议适合应用于协议开销要求较小的应用场合。

④ 综上所述,应用于工业控制领域的嵌入式系统在传输层建议采用传输控制协议(TCP 协议)。

【问题 3】(3 分)

这是一道要求读者比较嵌入式系统以太网接口的实现方法优缺点的简答题。本题的解答思路如下:

① 对于嵌入式处理器+网卡芯片的实现嵌入式系统增加以太网接口的方法(例如图 2-14),对嵌入式处理器没有特殊要求,只要把以太网芯片(例如:RTL8019AS, CS8900 等)连接到嵌入式处理器的总线上即可。此方法通用性强、不受处理器的限制,但是,处理器和网络数据交换通过外部总线(通常是并行总线)交换数据,速度慢,可靠性不高,且电

路板走线较复杂。

② 直接采用带有以太网接口的嵌入式处理器(例如 S3C4510B 等)的实现方法,要求嵌入式处理器面向网络应用而设计,带有 MII 等通用的网络接口。由于嵌入式处理器和网络数据交换通过内部总线完成,因此该实现方法的处理速度较快。

【问题 4】(3 分)

这是一道要求读者掌握嵌入式系统如何处理 IEEE 802.3 MAC 帧的综合理解题。本题的解答思路如下:

① IEEE 802.3 MAC 帧中包括帧前导码 PR、帧首定界符 SFD、目的地址 DA、源地址 SA、数据类型字段 TYPE、要发送的数据字段 DATA、填充字段 PAD 和帧校验序列 FCS 等 8 个字段。这 8 个字段中除了数据字段和填充字段之外,其余字段的长度都是固定的。

② 前导码字段 PR 占 7 个字节,每个字节的比特模式为 10101010,用于实现收发双方的时钟同步。帧起始定界符字段 SFD 占 1 个字节,其比特模式为 10101011,它紧跟在前导码后,用于指示一帧的开始。

③ 目的和源地址用于表示发送和接收帧的工作站的地址,各占 6 个字节。其中,目标地址可以是单播地址,也可以是多点传送地址或广播地址。

④ 以太网 MAC 帧的类型字段占 2 个字节,用于指定接收数据的高层协议。数据段就是网络层传送给数据链路层的 IP 包,其长度不能超过 1500 个字节。填充位用于保证以太网帧中数据段的最小长度不低于 46 个字节。

⑤ 帧校验序列(FCS)包含长度为 4 个字节的循环冗余校验值(CRC),由发送设备计算产生,在接收方被重新计算以确定帧在传送过程中是否被损坏。

⑥ 在图 2-14 电路原理图中通过 RTL8019 传输数据时,以太网 MAC 帧的帧前导码 PR、帧首定界符 SFD、填充字段 PAD 和帧校验序列 FCS 这四个字段都是由 RTL8019 芯片自动产生的,剩余的地址 DA、源地址 SA、数据类型字段 TYPE 和数据字段 DATA 等四个字段的內容需要上层软件配置处理。

⑦ 以太网 MAC 帧的传输顺序是从帧前导码的低位(即第 0 位)开始的,如图 2-18 所示,在网络中是以曼彻斯特编码比特流的形式出现的。

56位	8位	48位	48位	16位	46~1500字节	可选	32位
帧前导码	帧首定界符	目的地址	源地址	类型字段	数据段	填充位	帧校验码

0

图 2-18 IEEE 802.3 标准 MAC 帧结构

【问题 5】(3 分)

这是一道要求读者掌握嵌入式系统网络接口控制芯片复位方法及其物理地址存储方式的常识题。本题所涉及的知识点如下:

① 图 2-16 电路原理图中示意了网络接口控制芯片 RTL8019AS 的冷复位方法,即通过将其复位引脚 RSTDRV 置高电平进行的复位。RTL8019 的复位引脚 RSTDRV 是高电平有效的复位信号,高电平时间长度需大于 800 ns。通常在 RSTDRV 从高电平回到低电平的 100 ms 后,再对 RTL8019 进行读/写操作,以确保完全复位。这种复位方法的好处是当处理器复位时,以太网控制器也跟随其进行复位动作。

② 网络接口控制芯片 RTL8019AS 的另一种复位方法是通过读/写其复位端口 RST-PORT 来进行热复位。热复位方法能够保证网络接口控制芯片完全复位。RTL8019 共有 32 个输入/输出地址, 对应地址偏移量为 00h~1Fh。其中 18H~1FH 的 8 个地址为复位端口, 对该端口的偶数地址进行读或写入任何数, 都会引起以太网控制器的热复位。

③ 网络接口控制芯片的复位状态指它加电后所处的状态在该状态下中断状态寄存器的 RST 标志位(第 7 位, 地址为 07h, 位于第 0 页, 可直接读写)为 1, 此时网络接口控制芯片处于离线工作状态, 即不能发送或接收数据帧。在以太网控制器执行了正确复位之后该位为 1。一般在复位之后检查该标志位, 以确认是否复位成功。

④ 在完成对寄存器的初始化后, 还要对以太网控制器的物理地址(即 48 位的以太网控制器地址)进行设置。RAM 地址中的 0x0000~0x000B 这 12 字节(96 位)存储空间存放的是以太网控制器的物理地址。以太网控制器的物理地址本应该是 48 位(6 个字节)的, 因此将该物理地址在这 12 字节存储空间中按单双地址重复存储的。另外, 0x000B 后面的地址存储的是生产厂商的代码和产品标识代码, 这两个代码也是按单双地址重复存储的。

三、参考答案

表 2-29 给出了本案例问题 1 至问题 5 的参考答案, 供读者练习时参考, 以便查缺补漏。读者也可依照所给出的评分标准得出自己的测试分数, 从而大致评估自己对这一方面知识点的掌握程度。

表 2-29 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (4 分)	(1) 物理层 (1 分) (2) 数据链路层 (1 分) RTL8019AS 芯片完成 TCP/IP 协议簇中网络接口层的功能 (1 分) S3C44BOX 微处理器完成网络层、传输层、应用层的服务功能 (1 分)	
【问题 2】 (2 分)	宜采用传输控制 TCP 协议(1 分), 因为在工业控制领域的实时监控系统数据传输量并不大, 但可靠性要求较高。而 TCP 协议是一种面向连接、端对端的可靠通信协议, 它采用了诸如三次握手建立连接等多种机制来保证传输可靠性, 能弥补 IP 协议的不足(1 分)	
【问题 3】 (3 分)	① 嵌入式处理器+网卡芯片方法, 对嵌入式处理器没有特殊要求, 通用性强、不受处理器的限制, 但处理器和网络数据交换通过外部总线交换数据时速度慢、可靠性不高且电路板走线较复杂 (2 分) ② 带有以太网接口的嵌入式处理器要求带有 MII 等通用的网络接口, 处理器和网络数据交换通过内部总线完成, 处理速度快 (1 分)	
【问题 4】 (3 分)	从帧前导码的低位(即第 0 位)开始, 箭头方向参见图 2-18 (1 分) 帧前导码、帧首定界符、填充字段和帧校验序列这四个字段都是由 RTL8019 芯片自动产生的, 剩余的四个字段的内容需要上层软件配置处理(2 分)	
【问题 5】 (3 分)	(3) 复位引脚 RSTDRV 置高电平 (1 分) (4) 通过读/写其复位端口 RSTPORT 来进行热复位 (1 分) (5) 单双地址重复 (1 分)	

2.3 真题链接

本节讲述 2006 年下半年嵌入式系统设计师下午试卷试题 5。

一、试题描述

阅读以下关于嵌入式系统中异步串口的说明，回答问题 1 至问题 4。(15 分)

【说明】

异步串口是嵌入式处理器上最常用的资源之一。一般而言，异步传输的数据以帧的方式传输。每一帧有效数据前均有一个起始位，帧结束于一个或多个停止位。异步串口的数据由起始位和停止位分割成数据帧。常用的异步串口数据帧格式如图 2-19 所示。

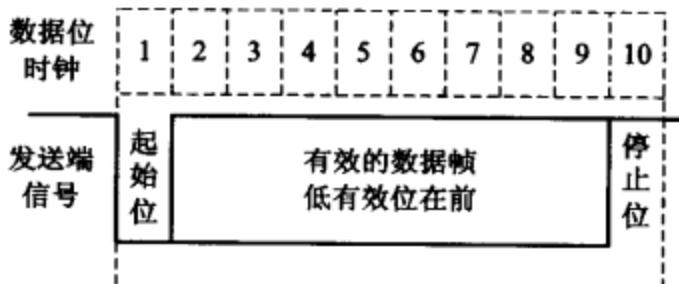


图 2-19 常用的异步串口数据帧格式

RS-232、RS-422 和 RS-485 都是常用的异步串口标准，它们的时序完全一样，只是在电气特性上有所区别，它们之间通过通用异步收发器(UART)可实现转换。

因为 UART 控制器可以集成到芯片中或者通过嵌入式处理器总线连接，所以，通常从 UART 发出的异步串口时序的逻辑电平都是处理器 I/O 电压标准(比如 TTL、LVTTTL 等标准)。若要求符合 RS-232、RS-422 或者 RS-485 的电气特性，则需要接口电路作转换。使用 MAX3232 芯片实现的串口电平转换的电路原理图如图 2-20 所示。

【问题 1】(3 分)

请说明异步传输和同步传输的不同之处。

【问题 2】(4 分)

根据图 2-19，请在 300 字以内简要描述异步串口的数据传输过程。

【问题 3】(4 分)

如果系统设计采用串行数据传输最高波特率为 115 200 b/s、16 分频，则时钟至少为多少赫兹？

【问题 4】(4 分)

图 2-20 所示的电路原理图设计实现了几路串口？每个串口为多少条连接线？

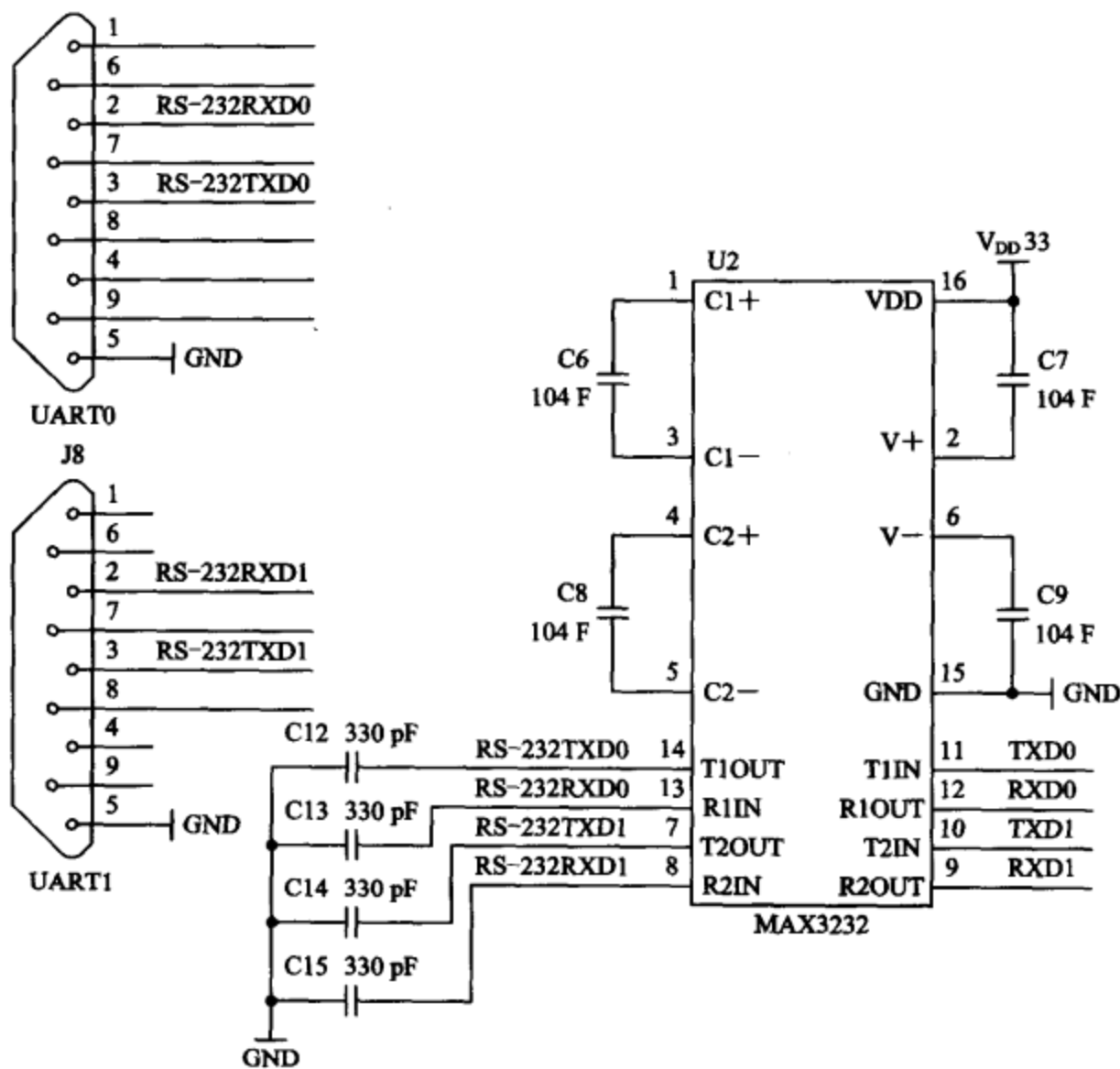


图 2-20 使用 MAX3232 芯片实现的串口电平转换的电路原理图

二、要点解析

【问题 1】(3 分)

异步传输与同步传输两者之间的主要区别在于发送器或接收器之一是否向对方发送时钟同步信号，其他区别见表 2-30。

表 2-30 异步传输与同步传输主要区别表

比较项	异步传输	同步传输
传输单位	字符	数据帧
同步信号	位形式	字符或位组合形
时钟同步信号	发送器和接收器具有相互独立的时钟，且两者中任何一方都不向对方提供时钟同步信号	由发送器或接收器提供用于同步的时钟信号
应用场合	键盘、RS-232 串口	计算机网络

异步传输的发送器和接收器具有相互独立的时钟(频率相差不能太多)，并且两者中任何一方都不向对方提供时钟同步信号。发送器与接收器双方在数据可以传输之前不需要协

调;发送器可以在任何时刻发送数据,而接收器必须随时都处于准备接收的状态。计算机主机与输入/输出设备之间一般采用异步传输方式,如键盘、典型的 RS-232 串口(用于计算机与调制解调器或 ASCII 码终端设备之间)。发送方可以在任何时刻发送一个字符(由一个开始位引导,然后连续发完该字符的各位,后面跟一个以上的停止位)。

同步传输可采用字符或位组合形式的帧同步信号(后者的传输效率和可靠性高),由发送器或接收器提供用于同步的时钟信号。在短距离的高速传输中,该时钟信号可由专门的时钟线路传输。计算机网络采用同步传输方式时,常将时钟同步信号植入数据帧(即帧前导码字段)中,以实现接收器与发送器的时钟同步。

【问题 2】(4 分)

本试题图 2-19 给出了在现有的嵌入式处理器上最常用的 1 个起始位、8 位有效数据、无奇偶校验位、1 个停止位的异步串口数据帧格式。有了起始位和停止位,可以使接收端与发送端保持帧同步。一帧数据是从最低有效位开始传输的。同时,要传输的数据中的每个字节要求至少 2 位,用于保证同步,因此,对于这种数据帧格式,同步位的开销比为 $(1+1)/(1+8+1)=20\%$ 。奇偶校验位指的是在有效数据帧传输完以后,接着再发送一个数据位,使有效数据帧中的各位数据加上此数据位的结果为奇数(或偶数)。

异步传输的数据是以帧的方式传输的。每一帧有效数据前有一个起始位,帧结束于一个或多个停止位。也就是说,为了保证同步(不至于因为一个位的丢失引起整个串行总线的混乱),异步串口的数据由起始位和停止位分割成数据帧。通常,每帧数据只有 8 位(或 5 位、6 位、7 位)。传输线在逻辑 1 时处于空闲状态。当每帧数据开始传输时,先发送一个起始位。起始位是从逻辑 1 到 0 的迁移,这个变化表明一帧数据开始传输,接收端则在开始检测到起始位的时候,按照收发两端事先约定好的通信速度检测后面的数据位,从而组成一帧数据。在传输的最后,利用一个(或多个)停止位(逻辑 1)使传输线回到空闲状态。然后发送方才可以发送下一个字节的数据。

【问题 3】(4 分)

如果系统设计采用串行数据传输最高波特率为 115 200 b/s、16 分频,则时钟至少为 $115\,200 \times 16 = 1.8432\text{ MHz}$ 。

【问题 4】(4 分)

在图 2-20 所示的电路原理图中,一片 MAX232 实现了两路的串口,分别为左上方的 UART0 和左下方的 UART1。对于 UART0 串口使用了 RS-232RXD0(接收数据线)、RS-232TXD0(发送数据线)和 GND(接地线)等三条连接线;对于 UART1 串口使用的三条连接线分别为 RS-232RXD1(接收数据线)、RS-232TXD1(发送数据线)和 GND(接地线)。

三、参考答案

【问题 1】(3 分)

异步通信系统的数据传输,其接收器时钟与发送器时钟是不同步的。

【问题 2】(4 分)

一帧数据是从最低有效位开始传输的。(1 分)

传输线在逻辑 1 时处于空闲状态。当每帧数据开始传输时,先发送一个起始位。起始

位是从逻辑 1 到 0 的迁移,这个变化表明一帧数据开始传输,接收端则在开始检测到起始位的时候,按照收发两端事先约定好的通信速度,检测后面的数据位,从而组成一帧数据。(2 分)

在传输的最后,利用一个停止位(逻辑 1)使传输线回到空闲状态。然后发送方才可以发送下一个字节的数据。(1 分)

【问题 3】(4 分)

时钟至少为 $115\,200 \times 16 = 1.8432\text{ MHz}$ 。

【问题 4】(4 分)

实现了两路的串口(2 分),每个串口为三条连接线(2 分)。



第3章 嵌入式软件程序设计

3.1 学习要点

3.1.1 考纲要求

根据考试大纲中相应的考核要求,在嵌入式系统软件设计方面,要求考生主要掌握以下几方面的内容。

一、嵌入式系统软件结构设计

二、嵌入式操作系统应用技术

1. 时间管理

- (1) 系统时间;
- (2) 时钟中断。

2. 内存管理

- (1) 静态内存管理;
- (2) 动态内存管理。

3. 任务管理和任务间的通信

- (1) 任务间的通信机制;
- (2) 信号量;
- (3) 邮箱;
- (4) 消息队列。

4. 异常处理

- (1) 异常处理方法;
- (2) 中断优先级处理方法;
- (3) 系统调用。

5. 嵌入式文件系统应用技术

6. 嵌入式系统图形用户接口(GUI)应用技术

7. 嵌入式系统数据库应用技术

三、嵌入式软件设计技术

1. 汇编语言设计

- (1) 数据类型;



- (2) 汇编语言程序结构;
- (3) 汇编语言程序设计及优化;
- (4) 子程序调用。

2. 嵌入式 C 语言设计

- (1) ANSI-C 的数据类型;
- (2) C 程序结构;
- (3) C 语言程序设计及优化;
- (4) 程序的编译与链接。

3. 面向对象程序设计与开发

- (1) 面向对象的分析与设计方法 UML;
- (2) 面向对象的编程语言;
- (3) 使用 C++ 进行嵌入式系统开发;
- (4) 使用 Java 进行嵌入式系统开发。

四、系统级软件设计技术

- 1. 嵌入式系统固件与系统初始化设计
- 2. 设备驱动程序设计
- 3. 硬件抽象层、板级支持包设计
- 4. 嵌入式软件的移植技术

五、协同设计

- 1. 软硬件任务分工和协调
- 2. 设计评审

六、嵌入式系统低功耗设计技术

- 1. 低功耗系统工作机制
- 2. 低功耗系统模型结构
- 3. 低功耗的软件设计技术

七、分布式嵌入系统设计

- 1. 分布式系统设计原理
- 2. 分布式系统的通信技术
- 3. 分布式系统设计应用

3.1.2 学习建议

从 2006 年下半年嵌入式系统设计师考试下午试卷的结构来看,本章是此次的考核重点,共有 3 道大题,每道大题 15 分,此部分的分值约占下午试卷总分的 60%。从 2006 年下半年下午试题的考试内容来看,在本章节知识点考核要求方面,考试难度中等,其中还有不少考题是跨章节综合考核的,因此建议读者在今后的备考过程中,要注意跨章节补充

有关的综合应用知识以弥补自己知识的盲点。嵌入式系统软件设计技术涉及的内容比较广,且可能考查到的知识点将越来越细,试题难度(主要体现在深度方面)也将随之增大。

熟练掌握一两门嵌入式系统软件设计的基本配置命令,是一名嵌入式系统设计师开展日常技术支持、故障维护等工作的基本技能。在解答这类试题的过程中,考生除了熟悉有关理论、方法和配置步骤之外,还需要具有丰富的实践配置经验。

建议考生一定要在嵌入式系统软件设计方面的知识点上多下功夫,尽可能地创造条件,在实际操作系统中熟悉典型的程序算法,并能多花力气去总结、比较,找异同点并抓规律,读懂源程序的相关输出信息、API 接口程序的编码技巧等。另外还需注意本书相关配置命令/语句的解释风格,这也可能是今后的一个出题方向。考生可以依照风格多下心思地解析典型的程序算法中每一条配置语句,努力做到熟能生巧,以便考试时能灵活变通,节约在这方面知识点的解题思考时间。本章也将力求以发展的眼光和实用的角度来预测、挖掘嵌入式软件程序设计的相关考核点,以增强考生学习相关知识点的目的性。

通过本章学习,可获得以下理论知识点及工程实践经验:

- ① 某水电站嵌入式安全监控系统任务实现技术;
- ② 嵌入式操作系统任务管理技术;
- ③ 嵌入式实时温度控制系统程序设计技术;
- ④ 某锅炉厂嵌入式安全监控系统软件需求及实现技术;
- ⑤ 嵌入式操作系统软件编码优化技术。

3.2 典型案例

3.2.1 案例1 任务实现技术

一、案例描述

阅读以下嵌入式系统任务的实现技术的相关说明,根据要求回答问题1至问题4。

(15分)

【说明】

某水电站根据安全监控的需要,决定采用嵌入式工程控制器(如图3-1所示)进行实时多任务安全监测。该系统有32路模拟量输入、96路开关量输入、16路脉冲量输入和水位落差量输入,产生显示信息、报警信号、监测报表和水位传感器控制信号等输出。



图 3-1 嵌入式工程控制器外观示意图

该水电站技术研发部门准备选用现有的工业控制系统及现成的各种模板等硬件，由本部门开发人员自行开发软件。所开发的软件主要包括：

(1) 应用程序，即按各类输入/输出量的采集、处理、显示、报警和制表等功能分别作为若干任务块，由这些任务组成应用程序。

(2) 多任务嵌入式实时操作系统，用来协调各任务的资源分配与管理，管理任务之间的联系与多任务操作。

在该嵌入式工程控制器系统分析阶段，参加该软件开发论证工作的多名系统设计师综合讨论后认为，本水电站安全监测系统有下列特点：

(1) 对各种开关、模拟或脉冲等量的监测，事实上只是周期性地重复执行相应的各个任务模块。

(2) 监视的对象比较固定，系统硬件的规模相对也固定不变。

(3) 尽可能允许把所有软件都固化在 EPROM 中。

(4) 人机交互作用的内容固定，即由操作人员通过键盘去调用此系统中所存入的相应应用程序，不会去修改应用程序。

根据以上安全监测系统的特点，技术研发人员郭工程师提出一个自行编制简化“多任务实时操作系统”的方案，以适应于协调管理硬件资源和监测对象的实际需求，其中只保留了标准的多任务实时操作系统的核心功能，例如 CPU 管理、中断管理和外设管理等。

【问题 1】(5 分)

设计简化的多任务嵌入式实时操作系统时，由于多个任务均可能要求占用 CPU 这个关键资源，因此 CPU 的任务管理是一个非常重要的设计内容。在本嵌入式实时操作系统中，任务作为占用资源的基本单位，总共有五个状态，分别是休眠状态、就绪状态、运行状态、等待或挂起状态和中断服务状态。在任何时候，一个任务只会处于其中的某一个状态。根据任务状态转换图(图 3-2)中已给出的信息，请将(1)~(5)空缺处的内容填写完整。

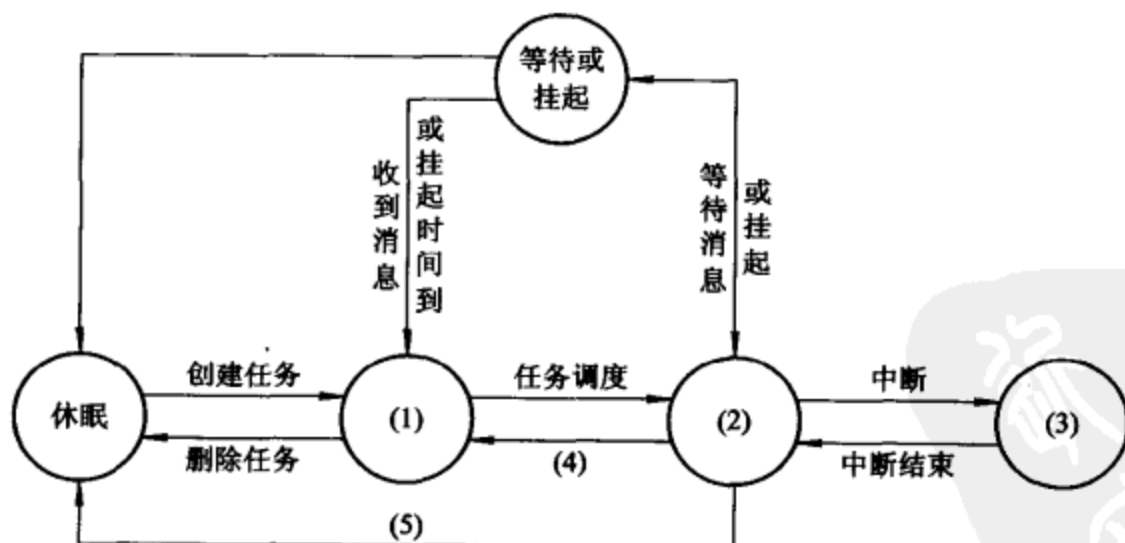


图 3-2 任务状态转换图

【问题 2】(3 分)

在嵌入式操作系统中，任务的创建主要有(6)和(7)两种可能的实现模型。在 $\mu\text{C}/\text{OS-II}$ 操作系统中，实时任务 mytask 创建过程的程序如图 3-3 所示。请在 100 字以内简要说明实时任务 mytask 创建过程的特点。

```

OS_STK Taskstk[1000];
void main( )
{INT8U err;
  OSInit();
  OSTaskCreate(MyTask, (void *)0, &Taskstk[999], &err);
  /* OSTaskCreate 函数用于创建一个新任务。参数:函数名,参数指针栈顶指针,
    优先级 */
  OSStart();
}

void mytask(void *pdata)
{for (;;)
  {do something;
   waiting;
   do something;
  }
}

```

图 3-3 $\mu\text{C}/\text{OS-II}$ 操作系统实时任务 mytask 创建过程的程序**【问题 3】(5 分)**

在该水电站安全监测系统的简化多任务嵌入式实时操作系统中，设计了一个“任务调度程序”，由它来分配 CPU 给相应的任务运行，即由“任务调度程序”从就绪队列中选出优先级最高的任务去获得 CPU 的使用权，该任务完成后把 CPU 的使用权再交还给“任务调度程序”。假设有 k 个任务 $T_1 \sim T_k$ ，在 RAM 区中设立一个“任务状态表”（其中存放每一任务的状态标志，这些任务标志由状态转换条件来改变），又在 ROM 区建立起一个“任务入口地址表”（其中存放每一任务块程序的入口地址）。

在“任务状态表”和“任务入口地址表”支持下，任务调度程序的执行遵循以下步骤：

- ① 任务调度程序搜索“(8)”队列，从中选出处于(9)态且具最高优先权的某个任务 T_i ；
- ② 任务调度程序通过映照关系从“(10)”中获得任务 T_i 所对应的任务块程序的入口地址；

- ③ (11)把 CPU 分配给任务 T_i 的程序运行；

- ④ T_i 任务块程序运行后，转换状态标志，释放(12)并通知“任务调度程序”。

【问题 4】(2 分)

假设一个任务正在 CPU 上运行，由于某种原因，系统决定调度另一个任务去运行，此时嵌入式实时操作系统将要进行一次任务切换。任务切换通常遵循以下基本步骤：

- ① 将处理器的运行上下文保存在当前任务的(13)中；
- ② 更新当前任务的状态，从运行状态变为就绪状态或等待状态；
- ③ 按照一定的策略，从所有处于就绪状态的任务中选择一个去运行；
- ④ 修改新任务的状态，从就绪状态变成运行状态；
- ⑤ 根据新任务的 TCB 的内容，恢复它的(14)。

二、要点解析

【问题 1】(5 分)

这是一道要求读者掌握嵌入式实时操作系统任务状态转换的基本常识题。本题的分析思路如下：

① 任务是相应任务块程序关于某数据集合的可并发的一次运行活动，即任务通常作为占有资源的最基本单位。通常在嵌入式实时操作系统中，任务总共有五个状态，分别是休眠状态、就绪状态、运行状态、等待或挂起状态和中断服务状态。在任何时候，一个任务只会处于其中的某一个状态。

② 休眠状态指任务还没有初始化或任务的执行已经结束。实际上对操作系统来说，该任务是不存在的。就绪状态是指任务已经具备了运行的条件，但是由于 CPU 正忙于运行其他任务，因此暂时不能运行的状态。

当一个任务被新创建时就处于就绪状态，处于就绪状态的任务可通过删除任务操作使任务处于休眠状态。根据图 3-2 中休眠状态与(1)空缺处的方向箭头，可判断(1)空缺处填写的内容是“就绪状态”。

③ 运行状态表示此任务正在占有 CPU，并在 CPU 上运行。处于就绪状态的任务只要把 CPU 控制权分配给它(任务调度)，它就能够立刻执行。

等待状态也叫挂起状态，是指任务因为正在等待某种事件的发生而暂时不能运行。例如，它正在等待某个 I/O 操作的完成，或者它同某个任务之间存在着同步关系，正在等待该任务给它发信号。此时，即使 CPU 已经空闲下来了，它也还是不能运行。处于挂起状态的任务可以被某些系统调用或事件激发而转到就绪状态。当一个任务在运行过程中需要等待某个事件时，将由运行状态转移到等待状态。

综合以上分析，可判断(2)空缺处填写的内容是“运行状态”。

④ 当有比正在运行的任务的优先级更高的事件发生时，运行状态将被高优先级的任务抢占，即高优先级中断可以使正在执行的任务转移到就绪状态。由此判断(3)空缺处填写的内容是“中断服务状态”；(4)空缺处可填入类似“任务被抢占”等内容。

⑤ 由于处于运行状态的任务完成后，可直接进入休眠状态，因此(5)空缺处可填入类似“删除任务”等内容。

图 3-4 给出了一个完整的任务状态转换图。

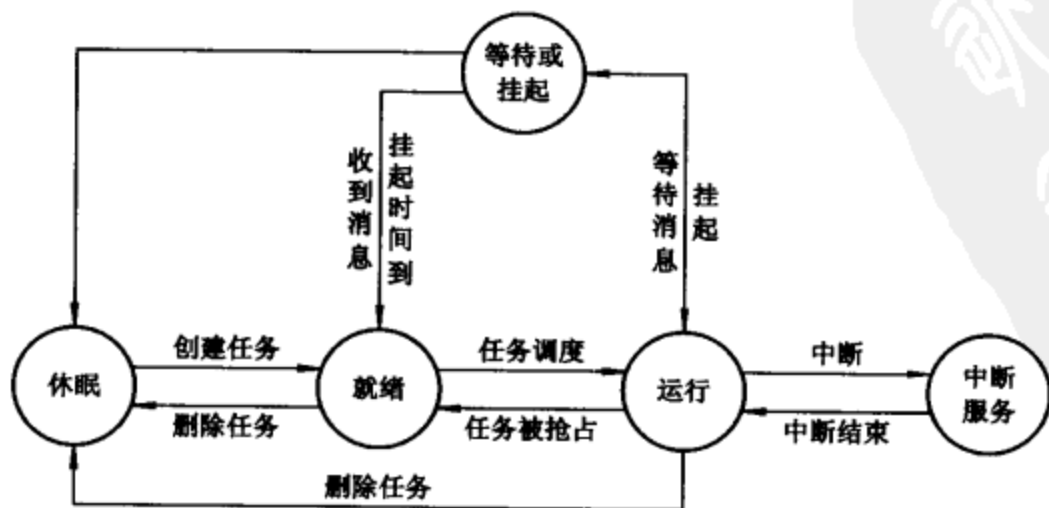


图 3-4 完整的任务状态转换图

【问题 2】(3 分)

这是一道要求读者掌握嵌入式实时操作系统任务创建的实现模型及其特点的分析理解题。本题的解答思路如下：

① 在一个嵌入式操作系统当中，任务的创建主要发生在系统初始化、用户提出请求和任务运行过程中等三种情况下。从技术的角度来说，实际上只有一种创建任务的方法，也就是在一个已经存在的任务当中，通过调用相应的系统函数来创建一个新的任务。

② 在嵌入式操作系统当中，任务的创建主要有 fork/exec 实现模型和 spawn 实现模型。其中，fork/exec 实现模型源于 IEEE/ISO POSIX 1003.1 标准，而 spawn 模型是从它派生出来的。这两种实现模型创建任务的共同特点是都要先为新任务分配相应的数据结构，存放它的各种管理信息，然后再为新任务分配内存空间，并存放任务的代码和数据。当这个新任务准备就绪后，就可以启动运行了。

③ 这两种实现模型的主要区别在于内存的分配方式，在 fork/exec 模型中，首先调用 fork 函数为新任务创建一份与父任务完全相同的内存空间，然后再调用 exec 函数装入新任务的代码，并覆盖原父任务的内容。这种实现模型的特点是具有继承的特性，即新建的子任务可以从父任务那里继承代码和数据等各种属性。图 3-5 给出了一个在嵌入式 Linux 操作系统中基于 fork/exec 模型创建某任务的实现过程。

```
void main( )
{
    int pid;
    pid=fork();
    if(pid>0)
        printf("parent task");
    else if(pid==0)
    {
        printf("child task");
        execvp("MyTask",NULL);
    }
}
```

图 3-5 嵌入式 Linux 操作系统中基于 fork/exec 模型创建某任务的实现过程

④ spawn 实现模型在创建新的任务时，直接为新任务分配一个全新的地址空间，然后再将其代码装入运行。在 $\mu\text{C}/\text{OS-II}$ 操作系统中，如图 3-3 所示试题中的 mytask 实时任务正是基于 spawn 实现模型创建的。

【问题 3】(5 分)

这是一道要求读者根据实际应用环境分析嵌入式实时操作系统任务调度程序主要工作步骤的分析题。本题的解答思路如下：

① 基于优先级的任务调度算法，任务的优先级取决于创建任务时赋予该任务的优先级，以及同一优先级的任务在就绪队列中的排队顺序。一个任务在创建时就被赋予了某一个优先级，但在运行过程中也可能被别的任务或自己通过系统调用赋予一个新的优先级。任务共有 256 个优先级(0~255)，其中 0 级是最高优先级，任务调度算法总是让处于就绪状态具有最高优先级的任务先执行。在同一优先级的就绪队列中按照先进先出算法选择一个任务运行，或者让同一优先级的任务按时间片分时运行。

② 在问题1要点分析中已得知,在图3-5中每一次中断服务例程(ISR)或者可能影响任务状态的系统调用后,实时操作系统都对系统中的任务重新进行一次调度,当更高优先级的任务就绪后,就发生任务切换,从而保证总是最关键的任务优先运行。

③ 根据试题中的关键信息“由任务调度程序从就绪队列中选出优先级最高的任务去获得CPU的使用权”和“任务状态表中存放每一任务的状态标志”可知,在“任务状态表”和“任务入口地址表”支持下,任务调度程序将首先搜索“任务状态表”队列,从中选出处于就绪态且具最高优先权的某个任务 T_i 。因此(8)空缺处应填入“任务状态表”,(9)空缺处所填入的内容是“就绪态”。

④ 根据试题中的关键信息“任务入口地址表中存放每一任务块程序的入口地址”可知,任务调度程序是通过映照关系从“任务入口地址表”中获得任务 T_i 所对应的任务块程序的入口地址。因此,(10)空缺处所填入的内容是“任务入口地址表”。

⑤ 根据试题中关键信息“设计了一个任务调度程序,由它来分配CPU给相应的任务运行”可知,任务调度程序具有分配CPU使用权的权限。因此,任务调度程序执行的第3步应由任务调度程序把CPU分配给任务 T_i 的程序运行,即(11)空缺处所填入的内容是“任务调度程序”。

⑥ 根据试题中关键信息“该任务完成后把CPU的使用权再交还给任务调度程序”可知, T_i 任务块程序运行后,转换相应的状态标志并记录于任务状态表中,接着释放CPU的使用权并通知任务调度程序。因此,(12)空缺处应填入“CPU的使用权”。

【问题4】(2分)

这是一道要求读者掌握嵌入式操作系统任务控制块和任务切换步骤的分析题。本题的解答思路如下:

① 任务控制块(TCB)是一个数据结构,用来保存该任务的相关参数,包括任务堆栈指针、状态、优先级、任务表位置、任务链表指针和CPU上下文信息等。其中,CPU上下文信息主要由各种CPU寄存器(包括通用寄存器、PC寄存器、程序状态字和核指针等)的当前值组成。在实际的嵌入式系统中,CPU上下文信息不一定直接存放在TCB当中,若存放在任务的栈当中,则可以通过相应的栈指针来访问。

② 可以把TCB看成是任务存在的惟一标志,用它来描述任务的基本情况以及它的运行变化过程。即当需要创建一个新的任务时,就为它生成一个TCB,并初始化这个TCB的内容。当需要终止一个任务时,只要回收它的TCB就可以了。而对于任务的组织和管理,也可以通过对它们的TCB的组织和管理来实现。

③ 由于某种原因,系统决定调度另一个任务去处于运行状态时,嵌入式实时操作系统将要进行一次任务切换,并把切换前处于运行状态的任务的运行上下文保存起来,且恢复新任务的上下文。因此处理器的运行上下文将保存在当前任务的TCB中,即(13)空缺处应填入“任务控制块(TCB)”。

④ 反之,当修改新任务的状态,从就绪状态变成运行状态后,操作系统将根据新任务的TCB的内容,恢复它的运行上下文环境,即(14)空缺处应填入“运行上下文环境”。

三、参考答案

表3-1给出了本案例问题1至问题4的参考答案,供读者练习时参考,以便查缺补

漏。读者也可依照所给出的评分标准得出自己的测试分数，从而大致评估自己对这一方面知识点的掌握程度。

表 3-1 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (5 分)	(1) 就绪状态 (2) 运行状态 (3) 中断服务状态 (4) 任务被抢占 (5) 删除任务 (每空 1 分)	
【问题 2】 (3 分)	(6) fork/exec (7) spawn (每空 1 分) 在创建新任务的时，直接为新任务分配一个全新的地址空间，然后再将其代码装入运行 (1 分)	
【问题 3】 (5 分)	(8) 任务状态表 (9) 就绪态 (10) 任务入口地址表 (11) 任务调度程序 (12) CPU 的使用权 (每空 1 分)	
【问题 4】 (2 分)	(13) 任务控制块(TCB) (14) 运行上下文环境 (每空 1 分)	

3.2.2 案例 2 任务管理技术

一、案例描述

阅读以下关于嵌入式操作系统任务的管理技术说明，根据要求回答问题 1 至问题 5。
(15 分)

【说明】

与传统的操作系统不同，在许多嵌入式操作系统当中，并没有使用“进程”或“线程”这两个术语，而是把能够独立运行的实体称为“任务(task)”。在某个嵌入式工程项目中，软件平台采用的是某一种实时的嵌入式操作系统。该项目有 1 个源文件 src1.c，如图 3-6 所示。

src1.c 文件中 taskSpawn 函数的功能是用来创建一个任务，其参数包括任务的名称、任务的优先级、任务所需要的栈空间的大小、任务的入口函数名以及这个函数的输入参数等。

【问题 1】(3 分)

嵌入式软件是实现嵌入式系统功能的关键。根据嵌入式软件的特点和实际应用中嵌入式系统的具体要求，在进行嵌入式软件平台选择时，通常要考虑哪些方面的因素？

【问题 2】(4 分)

在图 3-6 所示的源文件 src1.c 中，testTskA、testTskB 这两个任务分别完成什么功能？

```
int g_nSockId;
semId g_synSemId;
void testInit(void)
{
    创建 socket, 建立连接; /* g_nSockId 被赋值 */
    taskSpawn("tTestTskA", 50, 2000, testTskA, 0, ...);
    taskSpawn("tTestTskB", 50, 2000, testTskB, 0, ...);
}

void testTskA(void)
{
    char * pChRxBuf;
    pChRxBuf = malloc(100);
    while(1)
    {
        recv(g_nSockId, pChRxBuf, ...);
        ...
    }
}

void testTskB(void)
{
    char pChRxBuf[100] = "Send message back every 100 ms";
    while(1)
    {
        semTake(g_synSemId);
        send(g_nSockId, pChRxBuf, ...);
        ...
    }
}
```

图 3-6 源文件 src1.c

【问题 3】(3 分)

在图 3-6 所示的源文件 src1.c 中, 任务 testTskB 的 100 ms 定时功能由图 3-7 所示的源文件 src2.c 来实现。任务 testTskC 和任务 testTskB 之间通过信号量来进行任务间的同步。请将图 3-7 中(1)~(3)空缺处的内容填写完整。

【问题 4】(2 分)

在图 3-6 所示的源文件 src1.c 中, 变量 g_synSemId 存放在嵌入式系统内存地址空间的哪个位置? 在图 3-7 所示的源文件 src2.c 中, extern 语句完成什么功能?

【问题 5】(3 分)

在图 3-6 所示的源文件 src1.c 和图 3-7 所示的源文件 src2.c 中, testTskA、testTskB 和 testTskC 这三个任务是进程还是线程? 请在 200 字以内说明你的判断理由。

```
extern semId g_synSemId;
void test (void)
{
    创建同步信号量,并初始化为空; /* 即使用全局变量 g_synSemId */
    taskSpawn("tTestTskC",50,2000,testTskC,0,...);
}

void testTskC(void)
{
    while( (1) )
    {
        taskDelay( (2) );
        semGive( (3) );
    }
}
```

图 3-7 源文件 src2.c

二、要点解析**【问题 1】(3 分)**

这是一道要求读者掌握嵌入式系统软件平台选择的简答题。根据嵌入式软件的特点和实际应用中嵌入式系统的具体要求,在进行嵌入式软件平台选择时,通常要考虑的因素如下:

- ① 实际嵌入式系统应用项目对该硬件和软件平台的使用情况。
- ② 该平台的特殊需求,例如是否需要实时操作系统的支持。
- ③ 该平台是否有合适的开发工具。
- ④ 该平台所预留的资源或性能如何。
- ⑤ 该平台能否提供用户开发的设备驱动程序。
- ⑥ 该平台是否支持 HTTP、TCP、UDP 等 TCP/IP 协议簇的通信协议。
- ⑦ 提供该平台公司的发展目标是什么,是否有一个清晰的升级途径。
- ⑧ 可以使用该平台开发的开发人员有多少,对开发人员的培训费用是多少。
- ⑨ 该平台日后是否还需要完善,往往选择一个良好的平台比一个需要日后完善的平台要便宜得多。
- ⑩ 支持该硬件和软件平台公司的财务状况如何,能否提供日常的技术支持等。

【问题 2】(4 分)

这是一道要求读者掌握嵌入式系统 C 语言程序段分析的简答题。本题的解答思路如下:

① 在图 3-6 所示的源文件 `src1.c` 中, 首先定义了 2 个全局变量。它们分别是用于表示 socket 标识的 `g_nSockId` 和用于实现任务间同步的 `g_synSemId` 信号量标识。

② 接着定义了一个初始化函数 `testInit`, 在该函数中首先创建 socket, 并建立 socket 连接, 且对全局变量 `g_nSockId` 进行赋值, 然后调用 `taskSpawn` 函数创建任务 A 和任务 B。

③ 在“`void testTskA(void)`”函数(即任务 A 的入口函数)中, 首先定义了一个字符指针(`pChRxBuf`), 并用来动态申请 100 个字节的内存空间(`pChRxBuf = malloc(100)` 语句)。然后循环地从 socket 连接当中接收数据, 并保存在这个 100 个字节的内存缓冲区当中。

④ “`void testTskB(void)`”函数是任务 B 的入口函数, 它同样定义了一个字符指针(`pChRxBuf`)。用该指针来动态申请 100 个字节的内存缓冲空间, 并在这个缓冲空间中存入“Send message back every 100 ms”消息。然后循环地向 socket 连接发送这条响应消息。

通过以上分析, 可以得出 `testTskA` 任务是循环地从一个 socket 连接当中接收数据, 所接收的数据保存在字符指针(`* pChRxBuf`)所指向的 100 个字节内存缓冲区中。`testTskB` 任务实现每隔 100 ms 向 socket 连接发送一条响应消息的功能。

【问题 3】(3 分)

这是一道要求读者掌握嵌入式系统 C 语言定时程序编写的编程分析题。本题的解答思路如下:

① 在图 3-7 所示的源文件 `src2.c` 中, 首先用一个 `extern` 语句, 表明在这个 `src2.c` 源文件当中需要用到源文件 `src1.c` 当中的全局变量 `g_synSemId`, 即用于实现任务间同步的信号量标识。

② 然后在 `test` 函数中创建一个同步信号量, 并且把它初始化为空。接着调用 `taskSpawn` 函数创建任务 C。

③ “`void testTskC(void)`”函数是任务 C 的入口函数, 其功能在题目中已给出, 即循环地延迟 100 ms 的时间, 等到这 100 ms 时间过后, 就通过信号量 `g_synSemId` 去唤醒任务 B。

通过以上分析, 可以得出(1)空缺处应填入“1”, 用于实现无条件循环 `while` 所包含的程序段的功能。(2)空缺处应根据题目中要求“100 ms 定时功能”而相应填入“100”, 以完成延迟 100 ms 的时间并同时交出 CPU 资源的任务。(3)空缺处应根据题目中要求“任务 C 和任务 B 之间通过信号量来进行任务间的同步”, 而相应填入“`g_synSemId`”。

【问题 4】(2 分)

这是一道要求读者掌握嵌入式系统内存地址空间布局的简答题。本题的解答思路如下:

① 在实模式存储管理方案中, 嵌入式系统内存地址空间的布局一般可分为. `text` 代码段、. `data` 数据段、. `bss` 段、堆空间和栈空间这五个段。

② 在图 3-6 所示的源文件 `src1.c` 中, 全局变量 `g_nSockId` 和 `g_synSemId` 由于没有设置初始值, 因而存放在. `bss` 段中。

③ 在 `testInit` 函数中, 由于变量 `g_nSockId` 已被赋值, 因此该变量存放在. `data` 段中。

④ 在 testTskA 函数中, 字符指针 * pChRxBuf 因为是 testTskA 函数的局部变量, 所以存放在栈空间中。

⑤ 在 testTskA 函数中, malloc 函数分配的 100 字节的内存缓冲空间位于堆空间当中。

⑥ testInit、testTskA、testTskB 及 testTskC 等函数编译后的机器代码存放在 .text 段中。

⑦ 在图 3-7 所示的源文件 src2.c 中, extern 语句用于表明在这个源文件当中需要用到源文件 src1.c 当中的全局变量 g_synSemId, 即用于实现任务间同步的信号量标识。

【问题 5】(3 分)

这是一道要求读者分析在实际嵌入式系统环境中任务是进程还是线程的简答题。本题的分析思路如下:

① 进程是指一个正在运行的程序, 它包含相应的程序、CPU 上下文及一组系统资源等内容, 具有动态性、独立性和并发性等特性。

② 线程是指代码在进程的资源平台上的一条执行流程。在一个线程的运行过程中, 最直接、最必不可少的资源就是 CPU 上下文和栈, 这两种资源是不能与其他线程共享的, 每个线程都必须有自己独立的一份。

③ 从任务的创建过程来看, 在源文件 src1.c 和 src2.c 中 taskSpawn 函数所需要的参数主要是任务的优先级、所需要的栈空间的大小和入口函数名。可见, taskSpawn 函数创建的任务具有独立的优先级和栈空间, 这些都是创建一个线程所必需的资源, 而 CPU 上下文一般也是存放在栈空间当中的。

④ 在源文件 src1.c 和 src2.c 中, testTskA、testTskB、testTskC 这三个不同任务能够访问相同的全局变量。例如, testTskA、testTskB 这两个任务都使用了全局变量 g_nSockId; 任务 testTskB、testTskC 都使用了同步信号量 g_synSemId 这个全局变量。由此说明, 在这些任务之间可以很方便直接地去使用共享的内存单元, 而不需要经过系统内核来进行通信。而这正好就是线程的特点, 即同一个进程当中的所有线程可以共享该进程当中的各种资源, 包括内存地址空间和文件等资源。

通过以上分析, 可以得出在这个嵌入式操作系统中, 源文件 src1.c 和 src2.c 的 testTskA、testTskB 及 testTskC 这三个任务其实就是线程。然后根据题目不超过 200 字的要求, 整理出说明理由的答案: ① 从任务的创建过程来看, taskSpawn 函数创建的任务具有独立的优先级和栈空间, 这些都是创建一个线程所必需的资源, 而 CPU 上下文一般也是存放在栈空间当中的; ② 在本案例中, 对于不同任务, 它们能够访问相同的全局变量, 即在任务之间可以很方便直接地去使用共享的内存单元, 而不需要经过系统内核来进行通信。

三、参考答案

表 3-2 给出了本案例问题 1 至问题 5 的参考答案, 供读者练习时参考, 以便查缺补漏。读者也可依照所给出的评分标准得出自己的测试分数, 从而大致评估自己对这一方面知识点的掌握程度。

表 3-2 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (3 分)	① 实际嵌入式系统应用项目对该硬件和软件平台的使用情况 ② 该平台的特殊需求,例如是否需要实时操作系统的支持 ③ 该平台是否有合适的开发工具 ④ 该平台所预留的资源或性能如何 ⑤ 该平台能否提供用户开发的设备驱动程序 ⑥ 该平台是否支持 HTTP、TCP、UDP 等 TCP/IP 协议簇的通信协议 ⑦ 提供该平台公司的发展目标是什么,是否有一个清晰的升级途径 ⑧ 可以使用该平台开发的开发人员有多少,对开发人员的培训费用是多少 ⑨ 该平台日后是否还需要完善,往往选择一个良好的平台比一个需要日后完善的平台要便宜得多 ⑩ 支持该硬件和软件平台公司的财务状况如何,能否提供日常的技术支持 (能答出其中 6 点即可,每 2 小点 1 分,最高得 3 分)	
【问题 2】 (4 分)	testTskA 任务完成循环地从一个 socket 连接当中接收数据的功能,所接收的数据保存在字符指针(*pChRxBuf)所指向的 100 个字节内存缓冲区中(2 分) testTskB 任务完成每隔 100 ms 向 socket 连接发送一条响应消息的功能(2 分)	
【问题 3】 (3 分)	(1) 1 (2) 100 (3) g_synSemId(每空 1 分)	
【问题 4】 (2 分)	全局变量 g_synSemId 由于没有设置初始值,因此存放在 .bss 段中(1 分) extern 语句用于表明在这个源文件当中,需要用到源文件 src1.c 当中的全局变量 g_synSemId,即用于实现任务间同步的信号量标识(1 分)	
【问题 5】 (3 分)	源文件 src1.c 和 src2.c 的 testTskA、testTskB 及 testTskC 这三个任务其实就是线程(1 分)。理由主要有两点: ① 从任务的创建过程来看,taskSpawn 函数创建的任务具有独立的优先级和栈空间,这些都是创建一个线程所必需的资源,而 CPU 上下文一般也是存放在栈空间当中的(1 分) ② 在本案例中,对于不同任务,它们能够访问相同的全局变量,即在这些任务之间可以很方便直接地去使用共享的内存单元,而不需要经过系统内核来进行通信(1 分)	

3.2.3 案例3 程序设计技术

一、案例描述

阅读下列关于实时温控系统的软件设计的说明，回答问题1和问题2。(15分)

【说明】

某自动化系统工程研究所承担了一项嵌入式实时温度控制系统的研制任务。该计算机化温度自控装置由嵌入式实时控制部分与机电部分所组成。为了实时控制温度范围，机电部分采用了制冷压缩机、风机、加热器、指示灯和辅助灯源等各类温度调节用的机电设备。嵌入式实时温控装置硬件组成框图如图3-8所示。

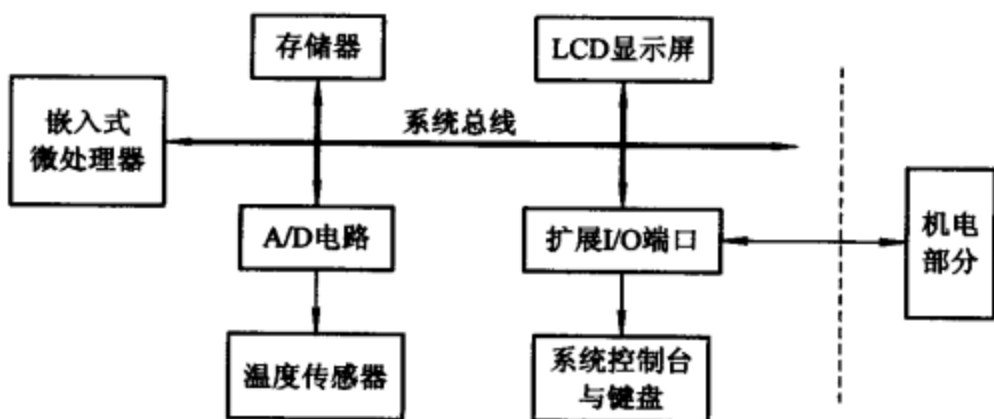


图3-8 嵌入式实时温控装置硬件组成框图

在该实时装置中，系统的使用人员可通过控制台及键盘进行功能的选择以及输入相应的参数或数据，来控制系统处于不同的工作状态，从而达到实时温度控制的目的。LCD显示设备用于反映系统当前的工作状态和机电部分工作状态，动态显现当前室内外温度等情况，便于由使用人员实时监控系统的工作。温度传感器实时地把室内外的温度值等转换为电压量，经A/D转换为数字量后，通过数据总线送入嵌入式微处理器，经微处理器运算处理后获得室内外温度值等相关参数。嵌入式微处理器和存储器等计算子系统将从事有关数据的处理与控制条件的分析计算，并产生相应控制输出用于实现对机电部分在各种条件下的启停控制等。

该装置可以对一个半封闭型的恒温室进行恒温控制和变温控制：

① 变温控制功能规定了高温温度HT和低温温度LT两个温度指标。当选定变温控制功能时，先使高温指示灯亮，并使室内温度迅速控制在 $HT \pm 6\%$ 范围以内。维持高温状态时间量12h后，高温指示灯熄灭，并使温度控制在 $LT \pm 6\%$ 范围以内。维持低温状态时间量也为12h，然后再使高温指示灯亮，进入高温HT状态，如此周而复始。

② 恒温控制功能只规定了一个恒温温度CT，且允许有 $\pm 5\%$ 范围变动，这时恒温指示灯亮，表示室内正处于恒温控制状态。

此类嵌入式实时温控系统的一个设计要点是应当力求避免机电部分(尤其是大功率的压缩机、加热器和风机等)频繁地开启与关闭，这样既可以延长机电部分的使用寿命，又可以降低能耗和提高系统控制的效率。为此，系统中温度传感采样时，采取了同时进行该恒温室内、外多点采样的方法，根据环境温度变化情况，相应地去改变制冷或加热的控制条件。

【问题 1】(5 分)

经系统分析后,初步拟订了如图 3-9 所示的系统控制程序流程。

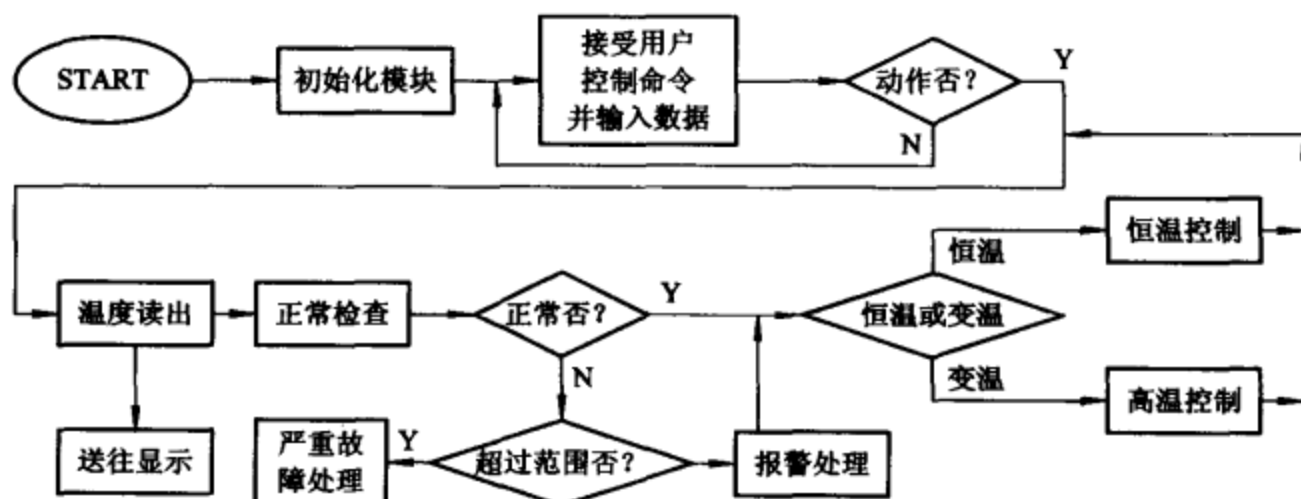


图 3-9 系统控制程序流程示意图

由此流程可知,该系统控制程序大体上由初始化模块、用户控制与输入模块、温度读出模块、正常检查模块与报警处理模块、变温与恒温控制模块以及时间量控制模块等组成。其中比较复杂的核心工作模块是变温控制或恒温控制模块的实现,各类有关中断的可靠实现将是系统实施有效控制的基础。

在这个系统控制程序设计时,初始化模块也占有相当重要的地位。请在 150 字以内简述初始化模块的基本内容。

【问题 2】(10 分)

某种传感器的输出值 Ratio 依赖于环境温度 $\text{temp}(-40^{\circ}\text{C} \leq \text{temp} \leq 50^{\circ}\text{C})$ 。对一组环境温度值(ITEMS 个),已经测量得到了相应的 Ratio 值(见表 3-3)。表 3-3 粗略地描述了曲线 $\text{Ratio}(\text{temp})$ 。

表 3-3 曲线 $\text{Ratio}(\text{temp})$ 的列表值

环境温度 temp	传感器的输出值 Ratio
-40°C	0.20
-20°C	0.60
-10°C	0.80
0°C	1.00
10°C	1.17
30°C	1.50
50°C	1.80

校正系数 K 是 Ratio 的倒数,因此也依赖于环境温度 temp。在数据处理中,需要用更多的列表值细致地描述曲线 $K(\text{temp})$,如表 3-4 所示。

表 3-4 曲线 $K(\text{temp})$ 的列表值

环境温度 temp	校正系数 K
-40℃	5.00
-39℃	4.55
-38℃	4.17
-37℃	3.85
⋮	⋮
-20℃	1.67
-19℃	1.61
⋮	⋮
-10℃	1.25
-9℃	1.22
⋮	⋮
50℃	0.56

在表 3-4 中, 各温度值所对应的 K 值是对表 3-3 进行线性插值再求倒数得到的, 具体的计算方法如下:

- ① 根据 temp 值, 在表 3-3 中用二分法查找;
- ② 若找到相应的温度值, 则按相应的 Ratio 值求倒数得到 K 值;
- ③ 若没找到相应的温度值, 则可确定 temp 所在的温度区间 $[Tp1, Tp2]$, 同时获得了相应的 Ratio1 和 Ratio2, 再按如下公式计算 K 值:

$$\text{Step} = \frac{\text{Ratio1} - \text{Ratio2}}{Tp1 - Tp2}$$

$$K = \frac{1.0}{\text{Ratio1} + \text{Step} \times (\text{temp} - Tp1)}$$

在程序中, 当 temp 高于 50℃ 或低于 -40℃ 时, 设定 $K=0$ 。

阅读以下 C 语言程序, 请将 (1)~(5) 空缺处填写完整。

【程序】

```
#include <stdio.h>

typedef struct{
    int Temp;           /* 环境温度 */
    double Ratio;       /* 传感器的输出值 */
}CURVE;
```



```

#define ITEMS 7
double GetK(int Temp, CURVE * p, int n)
{ /* 用二分法在 n 个元素的有序表 p 中查找与 Temp 对应的传感器输出值 */
    int low, high, m;
    double Step;
    low = 0;
    high = n-1;
    if ((Temp < p->Temp) || (Temp > (p+high)->Temp))
        return 0.0; /* 超出温度范围时返回 0.0 */
    while (low <= high)
    {
        m = (1) ;
        if (Temp == (p+m)->Temp)
            return (2) ;
        if (Temp < (p+m)->Temp)
            high = m-1;
        else
            low = (3) ;
    }
    p += high;
    Step = ((4) ) / ((p+1)->Temp - p->Temp);
    return 1.0 / (p->Ratio + Step * ((5) ));
}

void main()
{
    int Degree;
    double k;
    CURVE Curve[ITEMS] = {{-40, 0.2}, {-20, 0.60}, {-10, 0.8}, {0, 1.0},
                           {10, 1.17}, {30, 1.50}, {50, 1.8}};
    printf("环境温度校正系数\n");
    for(Degree = -40; Degree <= 50; Degree++)
    {
        k = GetK(Degree, Curve, ITEMS);
        printf("%3d%4.2f\n", Degree, k);
    }
}

```

二、要点解析

【问题 1】(5 分)

这是一道要求读者根据实际环境分析嵌入式实时温控系统初始化模块设计的程序分析题。本题的解答思路如下：

① 该系统控制程序中,初始化模块的作用是为系统运行做好必要的准备工作,一般包括运行初始环境的设置,将要运行的软硬件与受控设备进行起始工作状态的复位与参数设置等。

② 在系统开机启动时,必须先由系统对运行时将要占用的硬件资源(嵌入式微处理器、各类存储器、中断、DMA、I/O 端口与定时器等)进行初始状态的设置。这样在系统程序运行过程中,可方便地由系统去检测、比较各类占用资源的状态或进行各部件的正常性诊断检查,复位有关的硬件资源或部件。

③ 在系统开机启动时,还需对所控制的有关机电设备进行初始工作状态的复位或调整。同样在系统程序运行过程中,可方便地由相应的子程序去检测设备的状况或检查设备是否已达到正常的初态。

④ 由题干关键信息“系统中温度传感采样时,采取了同时采样该恒温室内、外多点采样的方法”可知,这些分布在恒温室内、外的温度传感器通常将通过不同的“中断”事件作用于该嵌入式实时温控系统,因此需要在初始化模块中设定中断的优先级别,以及初始化相关的中断机制。

⑤ 由试题关键信息“该系统控制程序大体上由初始化模块、用户控制与输入模块、温度读出模块、正常检查模块与报警处理模块、变温与恒温控制模块以及时间量控制模块等组成”以及图 3-9 可知,在初始化模块中需要对运行时控制软件将涉及到的有关程序模块的装入、调控参数和调控范围的默认值预设、实时数据库的初始化、其他关键参数的存取等进行初始设置。

⑥ 通常初始化模块程序设计时,还会在完成上述初始化工作后去显示出相应的系统提示信息,以方便把控制权交给用户,供用户进行选择性交互使用系统。

【问题 2】(10 分)

这是一道要求读者掌握线性插值计算及二分查找算法的 C 语言程序设计题。本题的分析思路如下:

① 试题中已给出函数 $\text{GetK}(\text{int Temp}, \text{CURVE} * p, \text{int } n)$ 用二分法在 n 个元素的有序表 p 中查找与 Temp 对应的传感器输出值,表中的元素已经按照温度有序排列。

② 结合本题的应用背景,二分查找算法是指先计算表的中间位置,即 $[(\text{low} + \text{high})/2]$,若待查元素等于中间位置上的元素,则查找成功并结束查找过程;若待查元素大,则在后半区间 $[(m+1, \text{high})]$ 继续进行二分查找;否则,在前半区间 $[(\text{low}, m-1)]$ 进行二分查找,如图 3-10 所示。本试题中 $\text{low}=0, m=3, \text{high}=6$ 。

0	1	2	3	4	5	6
-40℃	-20℃	-10℃	0℃	10℃	30℃	50℃
↑			↑			↑
low			m			high

图 3-10 二分查找算法示意图 1

③ 以查找温度值 20℃ 为例,由于 20℃ 高于 0℃,因此设置下一个查找区间为 $[(m+1, \text{high})]$,即 $[4, 6]$,如图 3-11 所示。

0	1	2	3	4	5	6
-40℃	-20℃	-10℃	0℃	10℃	30℃	50℃

low
m
high

图 3-11 二分查找算法示意图 2

④ 由于 20℃ 低于 30℃，因此取 high 等于 $m-1$ ，即下一个查找区间为 $([low, m-1])$ ，即 $[4, 5]$ 。再进一步分析，由于 20℃ 高于 10℃，因此取 low 等于 $m+1$ ，即再下一个查找区间为 $[5, 5]$ 。而查找区间 $[5, 5]$ 上的数据 $30 \neq 20$ ，至此可确定此次查找失败，即如图 3-12 所示。

0	1	2	3	4	5	6
-40℃	-20℃	-10℃	0℃	10℃	30℃	50℃

low
m
high

图 3-12 二分查找算法示意图 3

⑤ 由以上分析可知，(1)空缺处应填入计算表的中间位置的式子“ $(low+high)/2$ ”。

⑥ 程序中“`if(Temp==(p+m)->Temp)`”语句用于判断是否找到相应的温度值，若找到，则执行“`return (2)`”语句以返回相应温度的校正系数 K。由试题关键信息“若找到相应的温度值，则按相应的 Ratio 值求倒数得到 K 值”可推理出，(2)空缺处应填入“ $1.0/(p+m)->Ratio$ ”。

⑦ 当“`if(Temp==(p+m)->Temp)`”语句的判断条件不满足时，将执行“`if(Temp<(p+m)->Temp)`”判断语句。若待查元素小，将执行“`high=m-1;`”语句，即在前半区间 $([low, m-1])$ 进行二分查找。若待查元素大时，则在后半区间 $([m+1, high])$ 继续进行二分查找，因此 else 语句中则需将“low”指针向上移动，即将执行“`low=m+1`”语句。

⑧ 根据题干中给出的计算式子“ $Step = \frac{Ratio1 - Ratio2}{Tp1 - Tp2}$ ”，以及题目中要求线性插值再求倒数得到 K 值可推理出，(4)空缺处应填入“ $(p+1)->Ratio - p->Ratio$ ”。

⑨ 同理，根据题干中给出的计算式子“ $K = \frac{1.0}{Ratio + Step \times (temp - Tp1)}$ ”可推理出，(5)空缺处应填入“`Temp - p->Temp`”。

三、参考答案

表 3-5 给出了本案例问题 1 和问题 2 的参考答案，供读者练习时参考，以便查缺补漏。读者也可依照所给出的评分标准得出自己的测试分数，从而大致评估自己对这一方面知识点的掌握程度。

表 3-5 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (5 分)	① 对运行时将要占用的硬件资源进行初始状态的设置 ② 对所控制的有关机电设备进行初始工作状态的复位或调整 ③ 设定中断的优先级别以及初始化相关的中断机制 ④ 控制程序模块的装入以及相关参数的初始化设定 ⑤ 初始化后应提示把控制权转交给用户(每点 1 分)	
【问题 2】 (10 分)	(1) $(low + high) / 2$ (2) $1.0 / (p + m) \rightarrow Ratio$ (3) $m + 1$ (4) $(p + 1) \rightarrow Ratio - p \rightarrow Ratio$ (5) $Temp - p \rightarrow Temp$ (每空 2 分)	

3.2.4 案例 4 软件分析技术

一、案例描述

阅读以下关于嵌入式操作系统软件需求及实现的技术说明, 根据要求回答问题 1 至问题 5。(15 分)

【说明】

某锅炉厂根据安全监控的需要决定采用嵌入式实时控制系统进行实时多任务安全监测。该系统任务是按系统要求的固定的时间序列采集、处理和输出数据, 以实现对多台锅炉设备的综合控制。该实时控制系统经过对多个方案的对比、分析及讨论之后, 硬件计划采用主频 133 MHz 的 PowerPC603e 的 32 位微处理器、存储器、定时器、中断控制器、存储容量为 2048 KB 的双口存储器、15 路离散量接口、两路 A/D 转换器、一路 D/A 转换器和四路 422 半双工串口接口(其中两路的频率为 115 200 Hz, 另两路的频率为 38 400 Hz)组成。

该实时控制系统的软件需求如下:

① 系统要求以 10 ms 为周期从双口存储器中采集 2048 KB 的输入数据, 处理任务的时间约为 2 ms(简称为 10 ms 任务)。

② 系统要求以 20 ms 为周期, 从两路 422 接口(115 200 Hz)中采集已预先进行格式定义的大小为 128 B 的数据, 在完成处理(处理时间约为 4 ms)任务后, 分别输出大小为 16 B 的控制命令(简称为 20 ms 任务)。

③ 系统要求以 60 ms 为周期, 从两路 422 接口(38 400 Hz)中采集已预先进行格式定义的大小为 12 B 的数据, 从两路 A/D 接口采集 28 位数据。在完成处理(处理时间约为 3 ms)任务后, 分别输出大小为 4 B 的控制命令和一个 28 位 D/A 数据, 输出 10 路离散量控制数据(简称为 60 ms 任务)。

④ 作为系统的安全监控保障,系统要求在每 1 s 内对系统软硬件状态进行测试并完成系统工作的状态记录。该任务处理时间是 5 ms(简称为 1 s 任务)。

以上各周期任务的工作时序如图 3-13 所示。

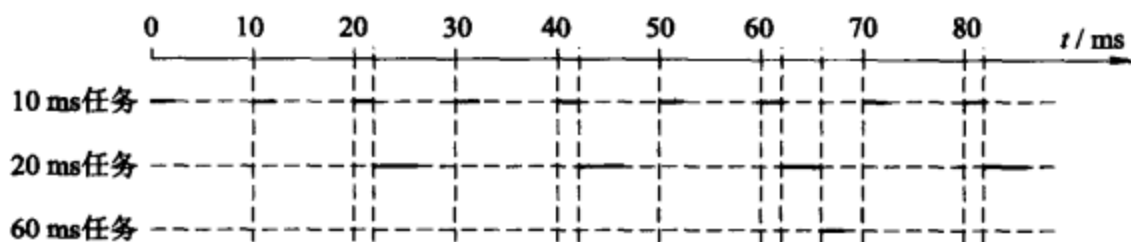


图 3-13 各周期任务的工作时序示意图

【问题 1】(4 分)

A 公司承担了该嵌入式实时控制系统的软件开发任务,其开发小组根据用户的硬件环境及软件的需求,就软件的实施方案展开了激烈的讨论。讨论的首要问题就是本系统中软件的运行平台是采用商用嵌入式实时操作系统还是在裸机上直接开发的方式。

张工程师(简称张工)提出为了保证系统的实时性,应该采用裸机上直接开发的方式,关键部分采用汇编语言编写。而郭工程师(简称郭工)提出,由于系统将由多种周期的处理任务组成,为了保证系统的可靠性,应该采用商用嵌入式实时操作系统作为本项目的开发基础。经过充分论证后,项目开发小组认为实时操作系统提供的任务调度功能,可以有效地管理本实时控制系统中 10 ms、20 ms 以及 60 ms 等多种周期任务的调度,并且实时操作系统提供的事件、信号和任务间的通信机制,可以有效地解决本系统中资源共享中的互斥问题。另外,采用商品化的软件,可以提高所开发软件的可靠性,还可简化软件开发成本,提高开发效率。因此项目开发小组采纳了郭工的建议。

根据本项目的要求,在接下来的嵌入式实时操作系统选型工作中,该开发小组重点要从哪些方面考虑操作系统的功能和性能?

【问题 2】(2 分)

请将以下(1)~(4)空缺处填写完整,完成关于该实时控制系统的工作时序关系的叙述。

根据本项目软件需求所提供的具体数据,由图 3-13 中 10 ms、20 ms 以及 60 ms 三种周期任务的工作时序可以获知,本项目软件系统的最小时间节拍定义为(1),系统最大工作周期为(2),在每个 10 ms 起点,(3)任务首先运行,占(4)时间,完成后转入处理被终止的任务;如果这个时刻时间到达 20 ms 的起点,则启动 20 ms 任务运行,以此类推,确保在 1 s 周期内所有任务按规定的时间序列执行。

【问题 3】(2 分)

根据图 3-13 中各周期任务的工作时序,说明所有任务的优先级排序以及本项目软件系统所采用的优先级分配策略。

【问题 4】(3 分)

根据该实时控制系统的软件需求,请设计并画出系统的处理流程图。

【问题 5】(4 分)

根据该实时控制系统的软件需求,简要说明项目硬件方案中双口存储器、离散量接口、A/D 接口、D/A 接口以及 422 接口应选择哪种数据输入/输出方法,并在 200 字以内

简述你的选择理由。

二、要点解析

【问题 1】(4 分)

这是一道要求读者根据实际环境进行嵌入式实时操作系统选型的分析试题。本题的解答思路如下:

① 根据问题要点,仔细阅读题干所提供的信息,并进行关键信息提炼。根据张工提出的方案,可以判断他认为在裸机上直接开发的系统实时性较好。而郭工则认为采用商用嵌入式实时操作系统开发的系统可靠性较好。

② 题目中只说明了经过充分论证后,开发小组最后采纳了郭工的建议,而没有给出拟采用的商用嵌入式实时操作系统的具体名称、性能指标。由此可间接推理,所采用的商用嵌入式实时操作系统既有好的可靠性,又能满足实时性要求。

③ 在进行软件设计时,通常要求所采用的嵌入式实时操作系统的系统时间开销,小于等于整个控制系统运行时间的总开销的 20%。

④ 进行软件系统时间性能评估时,重点考虑任务的上下文切换时间和中断响应时间。通过以上分析,可以得出选择嵌入式实时操作系统,主要考虑操作系统的下述功能和性能:

- ① 操作系统内核代码所占用的空间应比较小。
- ② 任务的上下文切换时间越短越好,中断响应时间越快越好。
- ③ 操作系统所提供的服务接口的开放性要好,接口类型应比较丰富,以利于灵活使用。
- ④ 操作系统对可靠性有较强的保障支持能力。
- ⑤ 使用简单,支撑环境配套好,可配置、可剪裁能力强。

【问题 2】(2 分)

这是一道要求读者根据实际应用环境分析嵌入式实时操作系统工作时序的试题。本题的解答思路如下:

① 根据问题要点,仔细阅读并找出题目所提供的关键信息。题干中关于软件需求的信息已将系统的任务按照执行的周期进行了分类,包括 10 ms 任务(即在 10 ms、20 ms 以及 30 ms 等时刻,该任务需要被唤醒执行)、20 ms 任务(即在 20 ms、40 ms 以及 60 ms 等时刻,该任务需要被唤醒执行)、60 ms 任务(即在 60 ms 和 120 ms 等时刻,该任务需要被唤醒执行)和 1 s 任务(系统软硬件测试任务)。

② 由于 10 ms 任务中处理任务的时间约为 2 ms,20 ms 任务中处理任务的时间约为 4 ms,60 ms 任务中处理任务的时间约为 3 ms,1 s 任务中处理任务的时间约为 5 ms,因此通过比较这些数据可知,系统最短的处理时间是 2 ms(10 ms 任务),最长的处理时间是 5 ms(1 s 任务)。

③ 通过以上分析,可以得到系统的最小时间节拍应定义为 10 ms,系统的主时间框架(即最大工作周期称)为 1 s。

④ 在每个 10 ms 起点,10 ms 任务首先运行,占 2 ms 时间,完成后转入处理被终止的任务;如果这个转入时刻的时间到达 20 ms 的起点,则启动 20 ms 任务运行;如果这个转

入时刻的时间到达 60 ms 的起点, 则启动 60 ms 任务运行; 以此类推, 确保在 1 s 周期内所有任务按规定的时间序列执行。试题中图 3-13 示出了 10 ms、20 ms、60 ms 这三种周期任务的工作时序。注意, “处理任务的时间”和“任务处理时间”所表达的意思有点不一样, 对于 1 s 任务系统, 只要求它每 1 s 内对系统软硬件状态进行测试并完成系统工作的状态记录工作即可, 并没有指定 5 ms 处理任务的时间具体在哪个时间段完成。可见这 5 ms 处理任务的时间的安排是灵活的, 只要插入到系统有空闲的时间段即可。

【问题 3】(2 分)

这是一道要求读者根据实际应用的工作时序图, 分析在保证完成各个实时任务的前提系统采用了什么任务调度方案。本题的解答思路可参考问题 2 的分析思路, 主要是抓住 10 ms 任务在 20 ms、40 ms、60 ms 以及 80 ms 时刻, 其处理任务的 2 ms 时间均优先安排在 20 ms 任务处理任务的 4 ms 时间之前。而 20 ms 任务在 60 ms 时刻, 其处理任务的 4 ms 时间优先安排在 60 ms 任务处理任务的 3 ms 时间之前。通过以上分析, 可以得出任务优先级从低到高排序是:

1 s 任务 60 ms 任务 20 ms 任务 10 ms 任务
最低 ----- > 最高

另外, 题目中已说明了 10 ms 任务、20 ms 任务和 60 ms 任务均是周期任务, 根据事件周期性的定义, 可以推理出本项目软件系统所采用的分配策略是小周期优先策略。

【问题 4】(3 分)

这是一道要求读者根据实际应用的设计系统的处理流程的分析题。本题的解答思路来源于题目所提供的关于“系统的软件需求”的文字说明, 以及图 3-13 中各周期任务的工作时序关系。最后得到的系统的处理流程如图 3-14 所示。

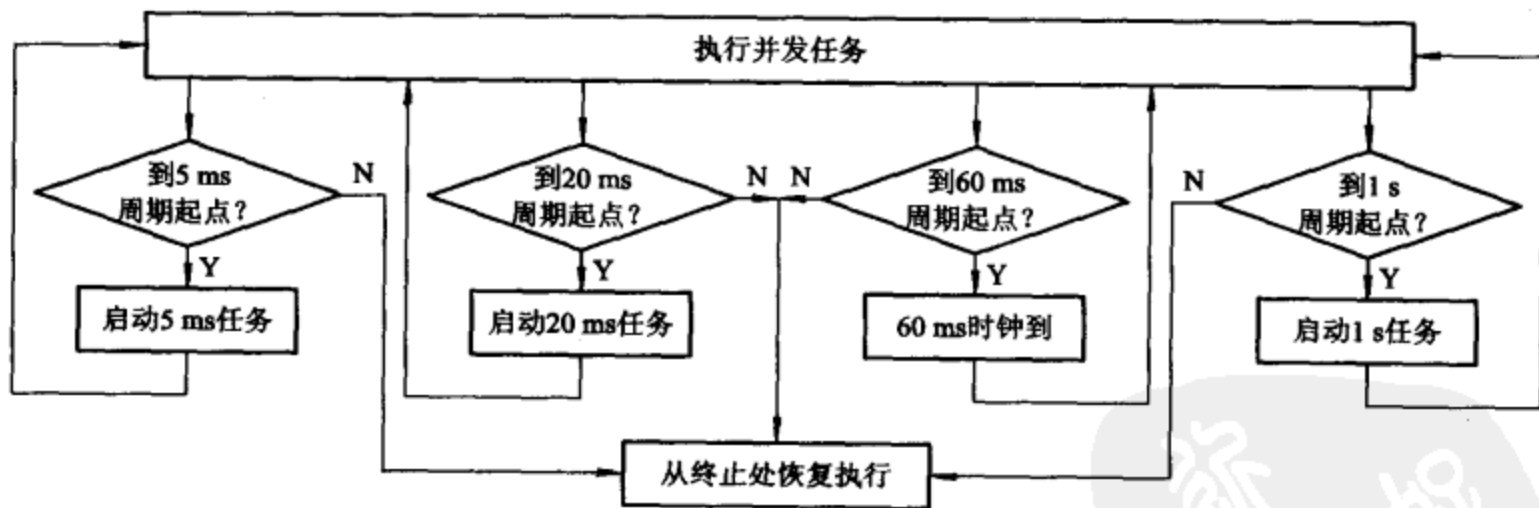


图 3-14 系统的处理流程图

【问题 5】(4 分)

这是一道要求读者根据应用环境并结合软件需求来选择硬件接口的数据输入/输出方式的分析题。本题的解答思路如下:

① 通常嵌入式系统设计中, 对于数据输入/输出处理方法有中断驱动和查询驱动两种方式。

② 双口存储器和离散量接口由于是直接访问存储器, 因此采用查询驱动方式较合适。

③ A/D 和 D/A 这两种接口存在数据的转换时间, 应在等待时交出处理机时间, 因此

应选用定时查询驱动方式。

④ 因为处理机速度远远比数据传输速度快,在采集时不易消耗太多的时间,而查询方式由于存在等待数据时间,消耗处理机时间较大,因此 RS-422 接口适合采用中断驱动方式。

三、参考答案

表 3-6 给出了本案例问题 1 至问题 5 的参考答案,供读者练习时参考,以便查缺补漏。读者也可依照所给出的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

表 3-6 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (4 分)	① 操作系统内核代码所占用的空间应比较小 ② 任务的上下文切换时间越短越好,中断响应时间越快越好 ③ 操作系统所提供的服务接口的开放性要好,接口类型应比较丰富,以利于灵活使用 ④ 操作系统对可靠性有较强的保障支持能力 ⑤ 使用简单,支撑环境配套好,可配置、可剪裁能力强 (答案类似即可,每小点 1 分,最高得 4 分)	
【问题 2】 (2 分)	(1) 10 ms (2) 1 s (3) 10 ms (4) 2 ms(每空 0.5 分)	
【问题 3】 (2 分)	任务优先级从高到低排序依次是: 10 ms 任务、20 ms 任务、60 ms 任务、1 s 任务(1 分) 优先级分配策略: 小周期优先策略(1 分)	
【问题 4】 (3 分)	5 ms 时钟到→启动 5 ms 任务→是否到 20 ms 周期起点?→是,则启动 20 ms 任务(否,则从终止处恢复执行)→是否到 60 ms 周期起点?→是,则启动 60 ms 任务(否,则从终止处恢复执行)→是否到 1 s 周期起点?→是,则启动 1 s 任务(否,则从终止处恢复执行)(3 分)	
【问题 5】 (4 分)	① 双口存储器和离散量接口由于是直接访问存储器,因此采用查询驱动方式较合适(1 分) ② A/D 和 D/A 这两种接口存在数据的转换时间,应在等待时交出处理机时间,因此选用定时查询驱动方式(1 分) ③ 因为处理机速度远远比数据传输速度快,在采集时不易消耗太多的时间(1 分),而查询方式由于存在等待数据时间,消耗处理机时间较大,因此 RS-422 接口适合采用中断驱动方式(1 分)	

3.2.5 案例 5 文件系统管理

一、案例描述

阅读以下关于嵌入式文件系统的技术说明, 根据要求回答问题 1 至问题 6。(15 分)

【说明】

嵌入式文件系统要为嵌入式系统的设计目的服务, 不同用途的嵌入式操作系统下的文件系统在许多方面各不相同。嵌入式文件系统的设计目标包括: 使用简单方便, 安全可靠, 实时响应, 接口标注的开放性和可移植性, 可伸缩性和可配置性, 开放的体系结构, 资源有效性, 功能完整性, 热插拔以及支持多种文件类型等。

文件的实现需要解决两个方面的问题: 一是如何来描述一个文件, 用什么样的数据结构来记录文件的各种管理信息; 二是如何来存储文件, 如何把文件的各个连续的逻辑块存放到磁盘的空闲物理块当中, 并记录逻辑块与物理块之间的映射关系。

带有文件分配表的链表结构是一种磁盘访问速度较快的文件物理结构。某嵌入式系统的文件分配表如图 3-15 所示。

文件目录项		文件1				0002			
块号	0	1	2	3	4	5	6	7	...
FAT表	0001	0002	0005	0004	FFFF	0006	0007	FFFF	...

图 3-15 某嵌入式系统的文件分配表

【问题 1】(2 分)

通常嵌入式文件系统不具有(1)功能。

- A. 提供建立、修改、改变和删除目录等服务
- B. 提供创建、打开、读写、关闭和撤销文件等服务
- C. 提供对文件的加密和解密功能
- D. 设置、修改对文件和目录的存取权限

【问题 2】(2 分)

文件的描述方法就是(1), 它是文件存在的标志, 是操作系统为了管理文件而设置的一种数据结构, 里面存放了与一个文件有关的所有管理信息。对于不同的操作系统, 它们的(1)所包含的内容是各不相同的, 通常主要包含文件的属性信息和(2)这两类信息。

【问题 3】(2 分)

在图 3-15 文件分配表中, 文件 1 总共有(3)个物理块。

- A. 3
- B. 4
- C. 5
- D. 无法确定

【问题 4】(3 分)

通常 FAT 文件系统由(4)、FAT 区、(5)、文件和目录区这四部分组成。在 FAT16 文件系统中, 若每个簇(cluster)的大小是 4 KB, 那么它所能表示的最大磁盘分区容量为(6)。

【问题 5】(4 分)

在如图 3-16 所示的树型文件中,方框表示目录,圆圈表示文件,“/”表示路径的分隔符,“/”路径之首表示根目录。

在图 3-16 中,假设当前目录是 A1,“COPY”表示拷贝命令,那么,将文件 f2 拷贝到根目录中的正确命令是(7);进程 P 打开文件 f1 的两种方式如下:

① fd1=open(" (8) /f1", o_RDONLY);

② fd1=open("/A1/Y1/f1", 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 的文件系统(9)。

下面对根目录中文件 f1 和子目录 Y1 中文件 f1 描述正确的是(10)。

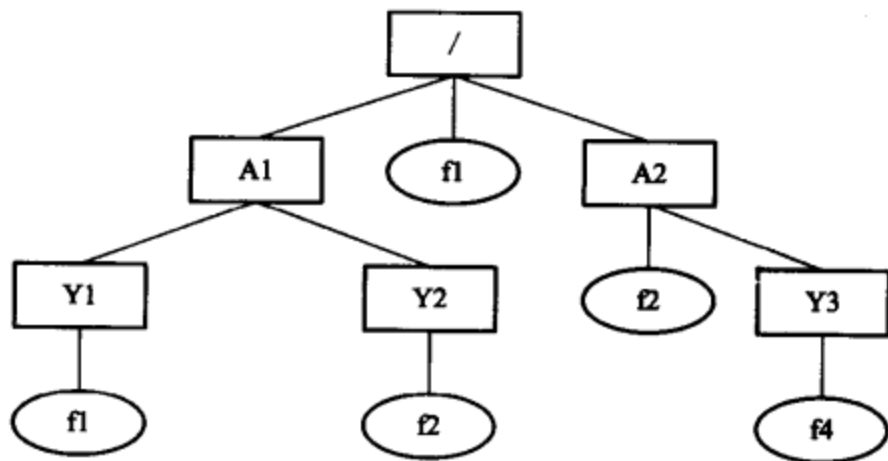


图 3-16 某文件系统结构图

- (7) A. COPY A1/Y2/f2/ B. COPY Y2/f2..
 C. COPY /Y2/f2..
 D. COPY /Y2/f2/
- (8) A. /A1/Y1 B. A1/Y1
 C. Y1 D. f1
- (9) A. 可以直接访问根目录下的文件 f1
 B. 可用从当前路径开始查找需要访问的文件 f1
 C. 只需要访问一次磁盘,就可以读取文件 f1,而方式 2 需要两次
 D. 只需要访问一次磁盘,就可以读取文件 f1,而方式 2 需要三次
- (10) A. 树型文件系统中不允许出现相同名字的文件
 B. 子目录 Y2 中文件 f2 和子目录 A2 中文件 f2 是不相同的
 C. 子目录 Y2 中文件 f2 和子目录 A2 中文件 f2 是完全相同的
 D. 子目录 Y2 中文件 f2 和子目录 A2 中文件 f2 是可能相同也可能不相同

【问题 6】(2 分)

影响文件系统可靠性的因素之一是文件系统的一致性问题,如果读取(11)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,另一个跟踪该块在空闲表中出现的次数。假设系统有 16 个块,检测程序通过检测发现表(12)状态下的文件系统是一致的。

- (11) A. 空闲表磁盘块 B. 用户目录文件
 C. 用户文件的某磁盘块 D. 系统目录文件
- (12)

选项	块 号	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A	计数器 1(使用中的块)	0	1	1	1	1	1	1	0	0	1	1	1	0	0	1	1
	计数器 2(空闲块)	1	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0
B	计数器 1(使用中的块)	0	1	0	1	1	1	1	0	0	1	1	1	0	0	1	1
	计数器 2(空闲块)	1	0	0	0	0	0	0	1	1	0	0	0	1	1	0	0
C	计数器 1(使用中的块)	0	1	0	1	1	2	1	0	0	1	1	1	0	0	1	1
	计数器 2(空闲块)	1	0	1	0	0	0	0	1	1	0	0	0	1	1	0	0
D	计数器 1(使用中的块)	0	1	0	1	1	1	1	0	0	1	1	1	0	0	1	1
	计数器 2(空闲块)	1	0	0	0	0	0	0	1	3	0	0	0	1	1	0	0

二、要点解析

【问题 1】(2 分)

这是一道要求读者掌握嵌入式文件系统基本功能的基本常识题。本题所涉及的知识点如下：

通常嵌入式文件系统比较简单，主要提供文件存储、检索和更新等功能，一般不提供保护和加密等安全机制。

嵌入式文件系统以系统调用和命令方式提供以下三种文件的操作服务：

- ① 提供建立、修改、改变和删除目录等服务；
- ② 提供创建、打开、读写、关闭和撤销文件等服务；
- ③ 设置、修改对文件和目录的存取权限。

通常嵌入式文件系统能支持 FAT32 等标准的文件系统，还支持自定义的实时文件系统，可支持多种存储设备，具有可裁减、可配置的特点。

【问题 2】(2 分)

这是一道要求读者掌握文件控制块定义及内容的基本常识题。本题所涉及的知识点如下：

① 文件的描述方法就是文件控制块(File Control Block, FCB)，它是文件存在的标志，是操作系统为了管理文件而设置的一种数据结构，里面存放了与一个文件有关的所有管理信息。

② 对于不同的操作系统，它们的文件控制块所包含的内容是各不相同的，通常主要包含文件的属性信息和存储信息这两类。

③ 文件的属性信息包括文件的类型和长度、文件的所有者、文件的访问权限、文件的创建时间、最后访问时间以及最后修改时间等。

④ 文件的存储信息是指文件在磁盘上的存放位置，即它被存放在哪一些物理块当中。

【问题 3】(2 分)

这是一道要求读者掌握文件分配表实现过程的综合分析题。本题的解答思路如下：

① 文件的物理结构研究的是如何把一个文件存放在磁盘等物理介质上，即以块为单位，研究如何把文件的各个连续的逻辑块存放在不同的物理块当中，从而建立逻辑块与物理块之间的映射关系。文件的物理结构主要有连续结构、链表结构和索引结构三种形式。

② 链表结构是把文件的各个逻辑块依次存放在若干个连续或不连续的物理块当中，然后在各个块之间通过指针连接起来，前一个物理块指向下一个物理块，从而形成一条链表。其优点是不存在存储碎片的问题，而且文件的大小也可以动态变化；缺点是在访问一个文件时，只能顺序地进行访问，如果想随机地访问，那么访问速度会比较慢。

③ 带有文件分配表的链表结构是在链表结构的基础上，把每一个物理块当中的链表指针抽取出来，单独组成一个表格（即文件分配表（File Allocation Table, FAT）），并把它存放在内存当中。如果要随机地去访问文件的第 n 个逻辑块，可以先从 FAT 表中查到相应的物理块地址，接着根据这个地址直接去访问磁盘，这样使得访问速度就比较快。

④ 由图 3-15 所示的文件分配表可看出，在整个文件系统中设置一个一维的线性表格，它的表项个数就等于磁盘上物理块的个数，并可按照物理块编号的顺序来建立索引。对于系统中的每一个文件，在它的文件控制块中记录了这个文件的第一个物理块的编号 X_1 ，然后在 FAT 表的第 X_1 项中，记录了该文件的第二个物理块编号 X_2 。依此类推从而形成了一个链表，在链表的最后一个节点中，存放了一个特殊的文件结束标识。

⑤ 在图 3-15 文件分配表中，通过文件 1 的目录项可知，它的第 1 个逻辑块存放在第 2 个物理块中，然后去查询 FAT 表可知，它的第 2、3、4 个逻辑块分别存放在第 5、6、7 个物理块中。第 7 个物理块中 $0xFFFF$ 是个特殊的值，表明文件的结束，所以该文件总共有四个物理块。

【问题 4】(3 分)

这是一道要求读者掌握 FAT 文件系统组成及其簇空间大小与最大磁盘分区容量关系的综合题。本题的分析思路如下：

① 通常 FAT 文件系统由保留区、FAT 区、根目录区、文件和目录区这四部分组成。其中，保留区（Reserved region）用于存放引导记录、引导代码、BIOS 参数块（BPB）、字节/扇区（512, 1024, 2048, 4096 等）、扇区/簇（1, 2, 4, 8, 16, 32, 64, 128 等）、根目录数、总扇区值以及介质类型（硬盘、软盘、Flash 等）。

② FAT 区存放有 FAT 信息列表（12/16/32 bits），每一个簇都有一个对应的 FAT 目录。根目录区存放目录信息列表（32 bytes）。文件和目录区存放簇信息。

③ FAT（File Allocation Table）是一种适合小卷集与双重引导需要的文件系统。采用 FAT 文件系统格式化的卷以簇的形式进行分配。默认的簇的大小由卷的大小决定。

④ 对于 FAT 文件系统，簇数目必须为 512 个字节到 65 536 个字节之间的 2 的次幂。表 3-7 列出了 FAT16 卷缺省的簇空间大小。

表 3-7 FAT16 文件系统默认的簇大小

卷空间大小/MB	每个簇的扇区	簇空间大小/B
0~32	1	512
33~64	2	1
65~128	4	2
129~255	8	4
256~511	16	8
512~1023	32	16
1024~2047	64	32
2048~4096	128	64

⑤ 查表可知,若每个簇的大小是 4 KB,那么它所能表示的最大磁盘分区容量为 255 MB。

【问题 5】(4 分)

这是一道要求读者掌握操作系统中文件系统的树型目录结构的综合分析题。本题的解答思路如下:

① 在图 3-16 所示的文件系统的树型目录结构中,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。

② 在树型目录结构中,从根目录到任何数据文件之间,只有惟一的通路。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惟一的。这样,可以解决文件重名问题。所以,第(10)题,虽然问题的研究对象是 f2 数据文件名,但由于其所在的子目录不同,因此子目录 Y2 中文件 f2 和子目录 A2 中文件 f2 是可能相同也可能不相同,正确答案为选项 D。

③ 从树根开始的路径名为绝对路径名。如果文件系统有很多级时,使用不是很方便,所以引入相对路径名,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。

假设当前目录是图 3-16 中的 A1,将文件 f2 拷贝到根目录中的正确命令是“COPY Y2/f2..”,其中“..”表示当前目录的上一级目录,恰好为根目录。因此,第(7)题的正确答案为 B。

④ 假设当前目录是 A1,进程 P 以“①fd1=open("Y1/f1", o_RDONLY);”或“②fd1=open("/A1/Y1/f1", o_RDONLY);”两种方式打开文件 f1。

对于方式 1,可用从当前路径开始查找需要访问的文件 f1,但需要访问两次磁盘,第一次查找目录文件 Y1,找到文件 f1 的物理位置,第二次便可以读取文件 f1。可见,从当前目录开始,采用相对路径名较之采用绝对路径名,可以减少访问目录文件的次数,工作效率较高。因此,第(8)题的正确答案为 C;第(9)题的正确答案为 B。

【问题 6】(2 分)

这是一道要求读者掌握文件系统的一致性问题的基本常识题。本题的分析思路如下:

① 影响文件系统可靠性的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(例如索引节点块、目录块或空闲管理块)的某磁盘块,修改后在写回磁盘前系统崩溃,则对系统的影响是不堪设想的。

② 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,其中,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。

③ 本案例中假设了系统有 16 个块,对于选项 A,若某个块的计数器 1 为状态“1”,则该块的计数器 2 的状态为“0”;若某个块的计数器 1 为状态“0”,则该块的计数器 2 的状态为“1”。这意味着该文件系统的一致性检查是正确的,因此(12)空缺处应选择 A。

④ 在(12)空缺处的供选择答案中,对于选项 B,块号为 2 的物理块在计数器 1 为状态“0”,这意味着没有文件使用这个块;但在该块的计数器 2 的状态也为“0”,这意味着这个块也不空闲。因此文件系统在一致性检查时将发现问题。

对于选项 C,块号为 5 的物理块在计数器 1 为状态“2”,这意味着这个块被重复分配了两次,文件系统在一致性检查时将发现问题。

对于选项 D,块号为 8 的物理块在计数器 2 为状态“3”,这意味着这个块被重复释放了三次,文件系统在一致性检查时将发现问题。

三、参考答案

表 3-8 给出了本案例问题 1 至问题 6 的参考答案,供读者练习时参考,以便查缺补漏。读者也可依照所给出的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

表 3-8 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (2 分)	C 或提供对文件的加密和解密功能(2 分)	
【问题 2】 (2 分)	(1) 文件控制块,或 FCB(1 分) (2) 存储信息(1 分)	
【问题 3】 (2 分)	B 或 4(2 分)	
【问题 4】 (3 分)	(4) 保留区 (5) 根目录区 (6) 255 MB(每空 1 分)	
【问题 5】 (4 分)	(7) B (8) C (9) B (10) D(每空 1 分)	
【问题 6】 (2 分)	(11) D 或系统目录文件(1 分) (12) A(1 分)	

3.3 真题链接

3.3.1 2006 年下半年嵌入式系统设计师下午试卷试题 2

一、试题描述

阅读以下说明和 X86 汇编语言代码,将应填入(n)处的字句写在答题纸的对应栏内。

【说明】

设有数组 X 和 Y。X 数组中有 X1、X2、…、X10 十个数，Y 中有 Y1、Y2、…、Y10 十个数。编制程序计算：

$$\begin{array}{llll} Z1 = X1 + Y1 & Z2 = X2 + Y2 & Z3 = X3 - Y3 & Z4 = X4 - Y4 \\ Z5 = X5 - Y5 & Z6 = X6 + Y6 & Z7 = X7 - Y7 & Z8 = X8 - Y8 \\ Z9 = X9 + Y9 & Z10 = X10 + Y10 & & \end{array}$$

设置一个 10 位的标志尺 LOGIC_RULE，若它的每一位决定了当前按顺序该做加法（该位为 0）或减法（该位为 1）操作，则由上列等式可知 LOGIC_RULE 中的低 10 位应是(1)。

【汇编程序代码】

```

dataarea segment                ; define data segment
    x dw  x1, x2, x3, x4, x5, x6, x7, x8, x9, x10
    y dw  y1, y2, y3, y4, y5, y6, y7, y8, y8, y10
    z dw  z1, z2, z3, z4, z5, z6, z7, z8, z9, z10
    logic_rule dw  (2)          ; LOGIC_RULE initialize
dataarea ends

program segment                 ; define code segment
main proc far
    assume cs:program, ds:dataarea
start:                          ; starting execution address
    (3)                          ; save old data segment
    sub ax, ax                   ; put zero in AX
    push ax                      ; save it on stack
; set DS register to current data segment
    mov ax, dataarea
    mov ds, ax

; MAIN PART OF PROGRAM GOES HERE
    mov bx, 0
    mov cx, 10
    mov dx, logic_rule
next:
    mov ax, x[bx]
    shr dx, 1
    jc subtract
    (4)
    jmp shortresult
subtract:
    sub ax, y[bx]
result:
    (5)                          ; result=>Zn
    Add bx, 2
    Loop next
    ret                          ; return to DOS

```

```
main endp          ; end of main part of program
program ends       ; end of code segment
endstart           ; end of assembly
```

附 指令及寄存器说明:

MOV	传送字或字节
PUSH	把字压入堆栈
SHR	逻辑右移
JC	有进位时转移
JMP	无条件转移指令
SUB	减法
ADD	加法
LOOP	CX 不为零时循环
JNE/JNZ	不等于时转移
CMP	比较

段寄存器 2 个:

CS	代码段寄存器
DS	数据段寄存器

数据寄存器 4 个:

AX	累加器
BX	基址寄存器
CX	计数寄存器
DX	数据寄存器

二、要点解析

(1) 由于 LOGIC_RULE 标志尺中某一位为 0 表示加法运算, 某一位为 1 表示减法运算, 因此根据题干中给出的 10 个计算式子“ $Z_1 = X_1 + Y_1$, $Z_2 = X_2 + Y_2$, $Z_3 = X_3 - Y_3$, $Z_4 = X_4 - Y_4$, $Z_5 = X_5 - Y_5$, $Z_6 = X_6 + Y_6$, $Z_7 = X_7 - Y_7$, $Z_8 = X_8 - Y_8$, $Z_9 = X_9 + Y_9$, $Z_{10} = X_{10} + Y_{10}$ ”可知, Z_1 的“+”运算对应于第 0 位(值为 0), Z_2 的“+”运算对应于第 1 位(值为 0), 依此类推直到 Z_{10} 的“+”运算对应于第 9 位(值为 0), 即 LOGIC_RULE 中的低 10 位二进制数为“0011011100”(注意, Z_{10} 的标志位在最左边, Z_1 的标志位在最右边)。

(2) 根据“LOGIC_RULE initialize”的解释信息可知, (2)空缺处所填写的语句是在数据段中对 LOGIC_RULE 标志尺初始化为(1)空缺处所表达的值, 其十六进制数为 00DCH (也可表达为 00dch, 或 220, 或 0b0011011100)。

(3) 根据“save old data segment”的解释信息可知, (3)空缺处所填写的语句是将把数据段压入堆栈, 以保存旧的数据段, 即使用“push ds”语句。

(4) 当执行“shr dx, 1”语句时, dx 是保存的 LOGIC_RULE 标志尺的值, 逻辑右移一位是 1 则执行“jc subtract”语句跳转到 subtract 执行减法运算, 否则顺序执行加法运算。因此在参照减法运算语句“sub ax, y[bx]”的基础上, 可推理出(4)空缺处所填写的语句是“add ax, y[bx]”。

(5) 根据“result=>Zn”的解释信息可知,(5)空缺处所填写的语句是“mov z[bx], ax”,用于将执行的结果移入 Z 存储区。

三、参考答案

(1) 0011011100 (3 分)

(2) 00dch, 或 00DCH, 或 220, 或等价的二进制形式 (3 分)

(3) push ds (3 分)

(4) add ax,y[bx] (3 分)

(5) mov z[bx],ax (3 分)

注: 参考答案中字母不区分大小写。

3.3.2 2006 年下半年嵌入式系统设计师下午试卷试题 3

一、试题描述

阅读以下说明和 C 语言代码, 回答问题 1 至问题 4, 将解答填入答题纸的对应栏内。
(15 分)

【说明】

有两个任务(编号分别为 0 和 1)需要访问同一个共享资源, 为了解决竞争条件(race condition)的问题, 我们需要实现一种互斥机制, 使得在任何时刻只能有一个任务访问该共享资源。代码一给出了一种实现方法。

【代码一】

```
1: intflag[2];      /* flag 数组, 初始化为 FALSE */
2: Enter_Critical_Section(int my_task_id,int other_task_id)
3: {
4:     while(flag[other_task_id]==TRUE);      /* 空循环语句 */
5:     flag[my_task_id]=TRUE;
6: }
7: Exit_Critical_Section(int my_task_id,int other_task_id)
8: {
9:     flag[my_task_id]=FALSE;
10: }
```

当一个任务要访问临界资源时, 就可以调用代码一给出的这两个函数。代码二给出了任务 0 的一个例子。

【代码二】

```
Enter_Critical_Section(0, 1);
.....使用这个资源.....
Exit_Critical_Section(0, 1);
.....做其他的事情.....
```

【问题 1】(2 分)

什么是临界资源(critical resource)? 请在 100 字以内简要说明。

【问题 2】(6 分)

代码一所示的方法能够实现共享资源的互斥访问吗？请在 300 字以内举例说明。

【问题 3】(4 分)

代码一采用了一种繁忙等待(busy waiting)的策略，这种策略的缺点是什么？请在 100 字以内字简要说明。

【问题 4】(3 分)

如果把 Enter_Critical_Section()函数中的两条语句互换一下位置，会出现什么情况？

二、要点解析**【问题 1】(2 分)**

请参见参考答案的说明。

【问题 2】(6 分)

请参见参考答案的说明。

【问题 3】(4 分)

本题考查的是任务之间的互斥问题，即基于繁忙等待(busy waiting)的任务互斥实现方法。其基本思路是，当一个任务要进入临界区，首先需要检查是否允许它进入，若允许，则直接进入；否则，则循环等待。

在多道程序系统中，各个任务是并发执行的，由于时钟中断的原因，使任务之间的执行顺序变得难以预测，每个任务都有可能在任意一条语句的后面被中断。在这种情形下，如果要采用基于繁忙等待的互斥实现方法，就必须考查所有的可能，即如果每个任务在不同的位置被中断时，能否正确地实现任务间互斥。

由于这种算法策略每个任务要循环地去判断当前能否访问临界资源，这样会浪费大量的 CPU 时间，而且如果设计不合理，容易导致死锁。

【问题 4】(3 分)

在“Enter_Critical_Section(int my_task_id, int other_task_id)”函数中，已提示“while (flag[other_task_id]==TRUE);”是一条空循环语句。如果将它调到“flag[my_task_id]=TRUE;”语句之后，将导致程序进入死锁状态。

三、参考答案**【问题 1】(2 分)**

在多道程序系统中，各进程可以共享各类资源，但有些资源一次只能供一个进程使用(1 分)。需要互斥访问的资源称为临界资源(critical resource)，如打印机、共享变量、表格等。(1 分)

【问题 2】(6 分)

代码一所示的方法不能实现资源的互斥访问。(2 分)

例如，考虑如下的情形：(4 分)

- ① 初始化时，flag 数组的两个元素值均为 FALSE；
- ② 任务 0 先执行，在执行 while 循环语句时，由于 flag[1]=FALSE，所以顺利结束，不会被卡住。假设这个时候来了一个时钟中断，打断它的运行；
- ③ 任务 1 先执行，在执行 while 循环语句时，由于 flag[0]=FALSE，所以顺利结束，

不会被卡住，然后就进入了临界区；

④ 以后，当任务 0 再执行时，也进入了临界区，这样就同时有两个任务在临界区。

【问题 3】(4 分)

缺点是会浪费大量的 CPU 时间。(2 分)

简要说明：繁忙等待(busy waiting)的基本思路是，当一个任务要进入临界区，首先需要检查是否允许它进入，若允许，则直接进入；否则，则循环等待，使用一个循环语句不断执行测试指令，从而浪费了大量的 CPU 资源。(2 分)

【问题 4】(3 分)

可能会出现死锁。(3 分)

3.3.3 2006 年下半年嵌入式系统设计师下午试卷试题 4

一、试题描述

阅读以下说明和 C 语言代码，回答问题 1 至问题 5，将解答填入答题纸的对应栏内。
(15 分)

【说明】

在实模式存储管理方案下，嵌入式系统的内存地址空间的布局一般可以分为五个段，分别是代码段(text)、数据段(data)、bss 段(bss)、堆(heap)和栈(stack)。

图 3-17 所示为一段例程。

```
1: #include <malloc.h>
2: unsigned char gvCh;
3: unsigned short gvShort;
4: unsigned int gvInt=0x12345678;
5: unsigned long gvLong=0x87654321;
6: void main(void)
7: {
8:     unsigned char array[10], * p;
9:     p=malloc(10 * sizeof(char));
10:    while(1);
11: }
```

图 3-17 某嵌入式系统例程段

【问题 1】(2 分)

代码段、数据段和 bss 段的大小是在什么时候确定的？

【问题 2】(3 分)

常用的堆空间的管理方法主要有哪两种？

【问题 3】(2 分)

当图 3-17 所示的程序经过编译链接并装入内存时，与 while 语句相对应的可执行代码存放在哪一个段？

【问题 4】(6 分)

在图 3-17 所示的程序中，变量 gvCh、gvShort、gvInt、gvLong、array 以及 p 分别存

放在哪一个段？

【问题 5】(2 分)

malloc 函数申请的空间位于哪一个段？

二、要点解析

本题是要求考生掌握任务的内存布局的分析理解题。通常，一个任务的内存布局分为五个部分：代码段、数据段、bss 段、堆和栈。代码段主要存放操作系统和应用程序的指令。例如，本试题与 while 语句相对应的可执行代码将存放在代码段中。

数据段主要存放操作系统和应用程序当中所有带有初始值的全局变量。例如，本试题有初始值的全局变量 gvInt 和 gvLong 将放在数据段中。

bss 段主要存放操作系统和应用程序当中所有未带初始值的全局变量。例如，本试题没有设置初始值的全局变量 gvCh 和 gvShort 将放在 bss 段中。

堆空间是动态分配的内存空间，在系统运行时，可以通过类似于 malloc/free 之类的函数来申请或释放一段连续的内存空间。

栈空间主要保存运行上下文以及函数调用时的局部变量和形参。例如，本试题的指针 p 和数组 array 都是 main 函数的局部变量，因此存放在栈当中。

对于代码段、数据段和 bss 段来说，它们的大小是固定的，在编译时即可确定。栈空间是由系统自动来维护的，而对于堆空间来说，为了提高存储管理的效率，减少开销，通常采用固定分区或可变分区存储管理方法。

三、参考答案

【问题 1】(2 分)

在编译时确定。(2 分)

【问题 2】(3 分)

固定分区(1 分)、可变分区。(2 分)

【问题 3】(2 分)

代码段。(2 分)

【问题 4】(6 分)

全局变量 gvCh 和 gvShort 由于没有设置初始值，所以放在 bss 段当中。(2 分)

全局变量 gvInt 和 gvLong 有初始值，所以放在 data 段当中。(2 分)

指针 p 和数组 array，它们都是 main 函数的局部变量，所以存放在栈当中。(2 分)

【问题 5】(2 分)

堆空间。(2 分)

第4章 嵌入式系统测试与维护

4.1 学习要点

4.1.1 考纲要求

根据下午考试大纲中相应的考核要求，在嵌入式系统测试与维护方面，要求考生主要掌握以下方面的内容：

一、系统测试

1. 测试环境
2. 测试计划(内容、方法、标准、过程、检验)
3. 硬件测试
4. 软件测试(单元测试、集成测试)
5. 软硬件联合测试
6. 实施测试

二、系统评估

三、软件维护

4.1.2 学习建议

在复习备考的过程中普遍存在着一种现象，就是读者们容易重视嵌入式系统软件程序编码和硬件接口电路的设计，而轻视嵌入式系统的测试和维护等知识点的复习。对于一个完整的嵌入式系统项目来说，必须配备专职的测试工程师，通过不断改善测试环境、改进测试技术来提高嵌入式系统的产品质量，并提升嵌入式系统开发公司的信誉，而不能简单地将测试工作交给开发人员来完成。

可以预见嵌入式系统硬件设备、软件程序进行优化和集成测试、嵌入式系统评估、软件维护等方面的知识点均可能成为今后下午试题考核的重点内容，且有可能占较大的分值比重。在解答这类试题的过程中，考生除了熟悉有关理论、方法和测试步骤之外，还需要具有丰富的实践经验。如果没有机会实践，则需要多阅读相关案例，尽量从这些案例中间接获取经验。本章也将力求以发展的眼光和实用的角度来预测、挖掘相关的考核点，以帮助读者在备考过程中分清主次、抓住重点，从而增强学习的目的性，减轻应试负担。

建议考生一定要在这方面的知识点上多下功夫，多阅读相关材料，多动笔做针对性的

练习,并从试题中多思考其出题的风格、出发点、形式等,直接获取应试经验,以便考试时能灵活变通并节约在这方面知识点的解题时间。

通过本章学习,可获得以下理论知识点及工程实践经验:

- ① 嵌入式软件单元测试技术;
- ② 嵌入式软件覆盖测试技术;
- ③ 嵌入式实时操作系统可移植性技术;
- ④ 嵌入式软件可靠性需求分析技术;
- ⑤ 嵌入式系统的性能评价技术。

4.2 典型案例

4.2.1 案例1 单元模块测试技术

一、案例描述

软件测试是从经济和效率的角度出发,对软件代码进行质量和正确性保证的一系列有计划的活动。阅读以下关于嵌入式软件单元测试的技术说明,根据要求回答问题1至问题4。(15分)

【说明】

某公司的技术开发小组经过一年的努力,编码完成了本公司嵌入式产品——宽带路由器的 NanOs 程序,该程序规模约为 31 200 行。公司经理指定郭工程师(以下简称为郭工)安排该测试工作。郭工决定由 A、B 两个测试小组每天独立测试同一个程序模块,并要求在每天测试中认真记录、统计有关的测试情况。A、B 两个测试小组由程序设计人员和有关用户代表两类人员构成,前 5 天对 NanOs 程序的测试有如表 4-1 所示的统计数据。

表 4-1 A、B 两个测试小组对 NanOs 程序测试的统计数据

日期	模块名	模块代码长度	A 组测试到的错误数	B 组测试到的错误数	两组检测相同的错误数
5.16	模块 1	1320	20	18	15
5.17	模块 2	1540	28	30	24
5.18	模块 3	1820	25	28	20
5.19	模块 4	2240	32	30	24
5.20	模块 5	2332	40	33	30

根据测试过程和结果数据,郭工提出了以下几条分析意见(或结论):

- ① 改正每天测试中 A、B 两个测试小组发现的共同错误,可以按线性速度减少 NanOs 程序中的错误数,即能稳妥地使 NanOs 程序中残留的错误数目越来越少。
- ② 经过每天认真测试,A、B 两个测试小组的测试技能将会越来越熟练,每个测试小组的测试效率将得到提高,因此,每天的测试进度会越来越快。
- ③ 由前 5 天的测试统计数据表明:由于每个测试小组都有用户代表参加,而用户代表

可以协助设计测试用例，测试的质量可以得到相当好的保证。

④ 由于测试用例的设计已适当考虑了故障的定位，目前已采用的设计方法是十分有效的。据此，郭工向公司经理表示可以在 15 天内完成 NanOs 程序的测试任务。

【问题 1】(3 分)

嵌入式软件的测试工作与普通台式机上应用软件的测试工作有着许多共同之处，但也有很大的区别，这些区别主要是由嵌入式软件的特点所造成的。请在 300 字以内简要说明与普通应用软件的测试工作相比，嵌入式软件的测试工作所具有的特点。

【问题 2】(3 分)

该公司的嵌入式产品——宽带路由器的软件系统会被划分为若干个较小的任务模块，由不同的程序员负责并同时进行编码。在各个模块编写完成且无编译错误后，在把它们集成起来之前，必须对各个模块分别进行测试。请在 150 字以内简要说明进行单元测试时主要从程序模块的哪些主要特征着手进行检查。

【问题 3】(5 分)

根据初期测试结果的统计数据来推算整个软件故障总数和测试进度是一个行之有效的软件测试管理办法。如果将 A、B 两个测试小组发现的错误数分别看成是两个“独立”事件，且认为前 5 天测试的 5 个程序模块在 NanOs 程序中具有相当好的代表性，请估算在正式进行测试前，NanOs 程序所具有的软件错误总数。请将表 4-2 中(1)~(5)空缺处填写完整，每个空缺处均需列出主要计算式子。

表 4-2 NanOs 程序错误总数计算过程

步骤	模块	本模块测试前故障总数估算	平均每一故障对应的代码行长度(行数)
①	模块 1	(1)	$\frac{1320}{24}=55$
	模块 2	(2)	(3)
	模块 3~模块 5 的计算数据略		
②	前 5 天平均每个故障对应的代码行数		(4)
③	整个 NanOs 程序测试前的故障总数		(5)

【问题 4】(4 分)

根据你的嵌入式软件测试经验，试判断郭工所提出的 4 条分析意见中哪几条是合理的，哪几条是存在问题的。请在 300 字以内简要说明你的理由。

二、要点解析

【问题 1】(3 分)

这是一道要求读者根据嵌入式软件的特点说明嵌入式软件测试工作区别于普通应用软件测试工作的分析题。本题的分析思路如下：

(1) 嵌入式软件的特点表现在以下几个方面：

- ① 嵌入式软件必须与嵌入式硬件一起对产品的故障负责。
- ② 嵌入式软件必须在很长一段时期内稳定的运行。

③ 嵌入式软件一般不会频繁地由用户进行升级。

④ 嵌入式软件可能使用在工厂的自动化控制设备等一些关键性的应用产品中。

⑤ 现实应用场合所发生的事件一般是异步且不可预测的，这就使模拟测试既困难又不可靠。

(2) 由于以上嵌入式软件的特点而造成其测试工作与普通应用软件的测试工作有着较大的区别，这些区别主要表现为：

① 在嵌入式系统中，由于在目标平台上通常没有很好的显示或输出能力，因此经常采用交叉编译和交叉测试的方法，常常需要在基于目标机的测试和基于宿主机的测试之间做出折中。

② 由于嵌入式环境的资源有限，CPU 主频低，内存小，只有少量的存储空间等；再者，嵌入式软件一般都固化在存储介质上，不易修改升级，且运行环境比较恶劣，因此在嵌入式软件测试时，除了验证逻辑上的正确性之外，还要看重系统的性能和健壮性。

③ 在嵌入式系统中，应用软件自身算法的复杂度和操作系统的任务调度，决定了系统资源的分配和消耗。因此，对嵌入式软件的实时性进行测试时，要借助一定的测试工具对应用程序的算法复杂度和操作系统的任务调度进行分析和测试。

④ 嵌入式系统的硬件一般采用专门的测试仪器进行测试，而这些测试工具一般不会在应用软件的开发中使用。

⑤ 由于嵌入式系统的开发是一个软硬件互相协调、互相反馈和互相测试的过程，因此在进行嵌入式软件测试时必须最大限度地模拟被测软件的实际运行环境，以保证测试的可靠性。但这种模拟有时是不太可靠的，这就增加了测试的难度，使得在系统测试时错误的定位较为困难。

【问题2】(3分)

这是一道要求读者掌握嵌入式软件单元测试从哪些方面着手测试的简答题。本题的解答思路如下：

(1) 单元测试也称为模块测试。一般来说，这个阶段的测试一般是在宿主机上进行的，因为宿主机上有丰富的资源和方便的调试环境。

(2) 单元测试一般采用的是白盒测试法，要尽可能地测试每一个函数、每一个条件分支、每一个程序语句，以提高代码测试的覆盖率。

(3) 嵌入式软件单元测试工作可以从模块接口、局部数据结构、重要的执行路径、出错处理和边界条件等程序模块特征进行着手检查。

① 重要的执行路径。在单元测试中，对路径的测试是最基本的任务。重点从以下几方面进行检查：算术运算的优先次序是否正确或理解错误，数据精度够不够，运算对象的类型是否匹配，逻辑运算符不正确或优先次序错误，循环终止条件不正确，分支循环的出口错误等。

② 局部数据结构。重点检查是否使用了尚未赋值或尚未初始化的变量，变量的初始值是否正确，变量名是否有错(如拼写错误)，变量的说明是否合适等常见的错误。

③ 模块接口。重点检查在模块调用或函数调用中，全局变量在各模块中的定义和用法是否一致，参数的使用是否正确，开/关语句、I/O 语句和文件的使用是否正确等，以保证测试模块的数据流可以正确地流入、流出。

④ 出错处理。好的设计应该能预测到出错的条件并且有出错处理的途径。虽然计算机可以显示出错信息的内容,但仍需要程序员对出错的模块进行处理,以保证其逻辑的正确性,便于用户进行维护。

⑤ 边界条件。边界条件的测试是单元测试的最后工作,也是容易出错的地方。

【问题 3】(5 分)

这是一道要求读者应用数学(逻辑或概率)知识估计软件代码错误总数的计算分析题。本题的主要解答思路如下:

(1) 题目中已给出了关键信息“将 A、B 两个测试小组发现的错误数分别看成是两个‘独立’事件”,设 N 为每个被测程序模块故障总数, N_A 为 A 组检测到的错误数, $P(A)$ 为 A 组检测到的错误数在被测程序模块故障总数中发生的概率, N_B 为 B 组检测到的错误数, $P(B)$ 为 B 组检测到的错误数在被测程序模块故障总数中发生的概率, N_{AB} 为 A、B 两组发现的相同错误数, $P(A \cap B)$ 为 A、B 两组发现的相同错误数发生的概率。根据事件 A、B 相互独立的充分必要条件 $P(A \cap B) = P(A)P(B)$, 即独立事件的 Bayes 概率计算公式, 由

$$P(A) = \frac{N_A}{N}, P(B) = \frac{N_B}{N}, P(A \cap B) = \frac{N_{AB}}{N}$$

得

$$\frac{N_{AB}}{N} = \frac{N_A}{N} \times \frac{N_B}{N}$$

即

$$N = \frac{N_A \times N_B}{N_{AB}}$$

(2) 根据以上计算公式, 可将表 4-1 所提供的每个程序模块的统计数据以图 4-1 的思路进行理解, 即表 4-1 中“两组检测相同的错误数”是指 A、B 两个测试小组独立发现的错误数的交集。因此可得出每个被测程序模块故障总数:

$$N = \frac{\text{A 组检测到的错误数 } N_A \times \text{B 组检测到的错误数 } N_B}{\text{A、B 两组发现的相同错误数 } N_{AB}}$$

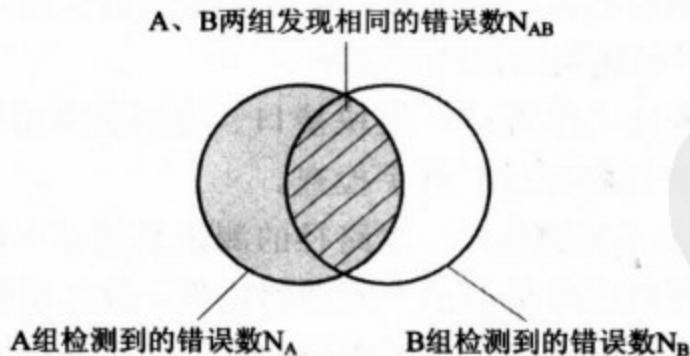


图 4-1 A、B 两个测试小组发现的错误数关系图

(3) 利用计算公式 $N = \frac{N_A \times N_B}{N_{AB}}$, 结合表 4-1 给出的前 5 天测试数据计算出每一个程序模块中故障总数的估算值, 并计算出每一个程序模块平均每一故障所对应的代码长度, 计算过程及结果见表 4-3。

表 4-3 每个程序模块测试前故障总数及平均每一故障所对应的代码长度

模块	模块测试前故障总数估算	平均每一故障对应的代码行长度(行数)
模块 1	$20 \times 18 / 15 = 24$	$1320 / 24 = 55$
模块 2	$28 \times 30 / 24 = 35$	$1540 / 35 = 44$
模块 3	$25 \times 28 / 20 = 35$	$1820 / 35 = 52$
模块 4	$32 \times 30 / 24 = 40$	$2240 / 40 = 56$
模块 5	$40 \times 33 / 30 = 44$	$2332 / 44 = 53$

(4) 计算出前 5 天平均每个故障对应的代码行行数, 即 $(55 + 44 + 52 + 56 + 53) / 5 = 52$ 行/每个故障。

(5) 依题意“前 5 天测试的 5 个程序模块在 NanOs 程序中具有相当好的代表性”, 从而推算出在正式进行测试前, NanOs 程序所具有的软件错误总数, 即 $31\,200 / 52 = 600$ 个。

(6) 顺便指出, 由于目前所进行的仅仅是模块测试, 即部件级测试, 还没有考虑到集成测试和系统测试中将必然会出现的 NanOs 程序错误数量。可以预知总的 NanOs 程序错误数必然是大于 600 个, 这是相当高的错误率。

【问题 4】(4 分)

这是一道要求读者根据具体应用项目分析嵌入式软件测试工作组织原则和管理原则的分析题。本题的主要解答思路如下:

(1) 通常, 嵌入式软件测试工作包含有系统平台测试、单元测试、集成测试、系统测试和测试结果分析五个步骤。对于本题所提及的只是进行了宽带路由器 NanOs 程序的模块级测试工作, 除此之外还需进行模块之间的集成测试和系统级测试。一般说来, 部件测试是属于基础性的测试工作, 也是相对容易的测试工作。模块之间的集成测试和系统测试的组织管理要更困难些, 也往往会发生纠正一个错误将可能耗费更多的人力和机时。郭工程师没能强调指出这一主要的测试局部性问题。由以上分析可知, 郭工所提出的第 2 条分析意见, 即“每天的测试进度会越来越快”是过于乐观的。

(2) 在本题表 4-1 中统计所列出的仅是检测到的错误, 尽管试题描述中说明了已去改正 A、B 两组检测到的相同错误。改正了这些共同发现的错误后虽然可以减少错误, 但尚未改正的错误和潜在的错误数量仍占很大的比例(第 1 天 $24 - 15 = 9$ 个, 第 2 天 $35 - 24 = 11$ 个, 第 3 天 $35 - 20 = 15$ 个, 第 4 天 $40 - 24 = 16$ 个, 第 5 天 $44 - 30 = 14$ 个), 本例中前 5 天共有 $9 + 11 + 15 + 16 + 14 = 65$ 个错误没有改正, 约占到已被测试模块估计错误总数的 $36.52\% \left(\frac{9+11+15+16+14}{24+35+35+40+44} = \frac{65}{178} = 36.52\% \right)$, 并且这些错误还需进行回归测试等工作。对于那些未改正的及未被发现的错误, 往往更难测试到, 需要花费更多的人力或时间才能改正它们。可见错误数不可能线性减少, 因此郭工所提出的第 1 条分析意见是错误的。

(3) 改正错误时还可能会引发新的错误。统计发现, 某些局部性错误的改正往往在集成测试和系统测试时暴露出新的问题。另外, 测试人员的测试技能不是一朝一夕或在短期内即可以形成的, 这一点也说明郭工所提出的第 2 条分析意见是片面的。

(4) 对于郭工所提出的第 3 条分析意见存在着组织测试工作方面的问题。通常测试工

续表

问题与分值	参考答案及评分标准	自评分
【问题4】 (4分)	<p>郭工程师的4条分析意见基本上都是很片面的(1分)。主要理由有:</p> <p>① 除了模块部件级测试外,还需要进行集成测试和系统测试。</p> <p>② 改正共同发现的错误虽然可以减少错误,但本案例中前5天共有65个错误没有改正,约占到已被测试模块估计错误总数的36.52%,且这些错误需进行回归测试。那些未能发现的错误需要花费更多的人力或时间才能被测试到。故错误数不可能线性减少。</p> <p>③ 改正错误时还会引发新的错误。统计发现,某些局部性错误的改正往往在集成测试或系统测试时暴露出新的问题。</p> <p>④ 测试工作必须由有资历的专职测试人员组成,原编程人员原则上不进入测试小组;另外还需完善制定测试计划等工作。</p> <p>⑤ 本例所用的测试定量分析方法过于粗糙,确定的时间进度可信度差。</p> <p>(答案类似即可,每答对1条理由得0.5分,答出5条理由得3分)</p>	

4.2.2 案例2 覆盖测试技术

一、案例描述

阅读以下关于嵌入式软件覆盖测试的技术说明,根据试题要求回答问题1至问题5。(15分)

【说明】

某嵌入式产品开发公司在研制与开发各类嵌入式应用软件的过程中,深切地体会到“软件测试”的重要性与复杂性,认为这是关系到公司信誉、软件质量和软件维护的关键技术活动之一。公司的林总工程师多次召集公司有关的管理人员与技术骨干,分析软件测试的规范化问题,讨论中一致认为规范化应涉及下列一些基本的软件测试活动:

- ① 成立一个或多个独立的测试小组;
- ② 拟定测试计划,进行测试项目配置管理;
- ③ 编制测试大纲;
- ④ 设计并生成各类测试用例;
- ⑤ 以一系列“测试小周期”实施软件测试;
- ⑥ 确认问题并产生相应的软件问题报告单;
- ⑦ 在程序设计人员修改完成后,进行回归测试。
- ⑧ 编写测试报告,并分析测试结果。

由于嵌入式软件的开发与通用软件的不同点在于需要采用交叉开发的方式,即开发工具运行在软硬件配置丰富的宿主机上,而嵌入式应用程序运行在软硬件资源相对缺乏的目标机上,因此在嵌入式软件的测试工作中,测试工具的运行往往在宿主机上,而测试所需要的信息却在目标机上产生,需要通过一定的物理/逻辑连接传输到宿主机上,并由相应的测试工具接收。图4-2示意了嵌入式软件覆盖测试的基本原理。

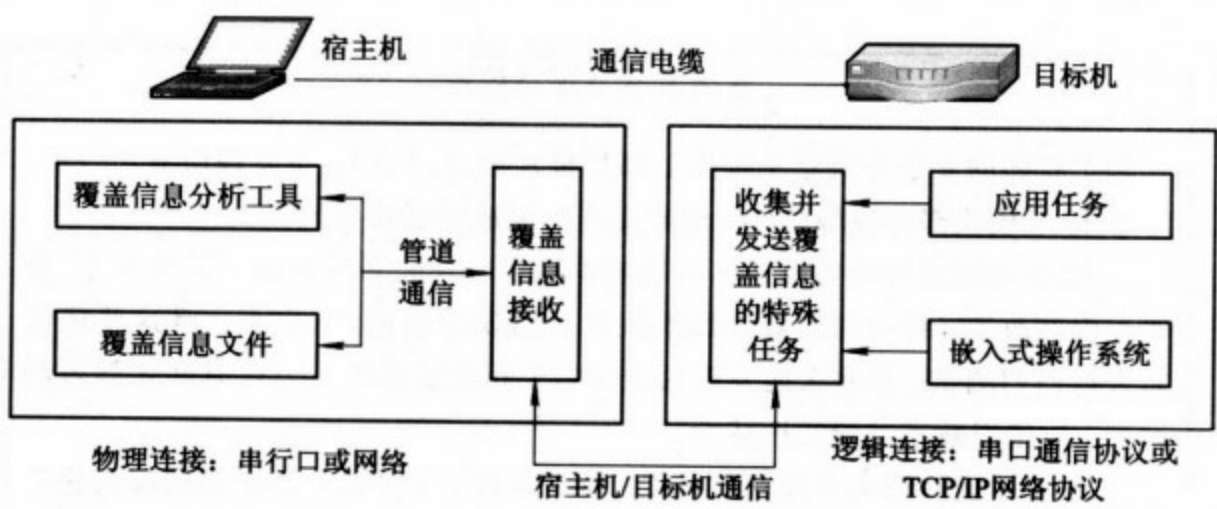


图 4-2 嵌入式软件覆盖测试的基本原理图

【问题 1】(3 分)

嵌入式软件测试是从经济效率的角度出发，对软件代码进行质量和正确性保证的一个过程，该测试过程是软件开发中的一个重要环节，也是软件从开发过程到应用过程的关键一环。图 4-3 给出了嵌入式软件测试的统一测试模型。请结合该公司总结出的规范化测试过程所包含的基本测试活动，将图 4-3 中(1)~(3)空缺处填写完整。

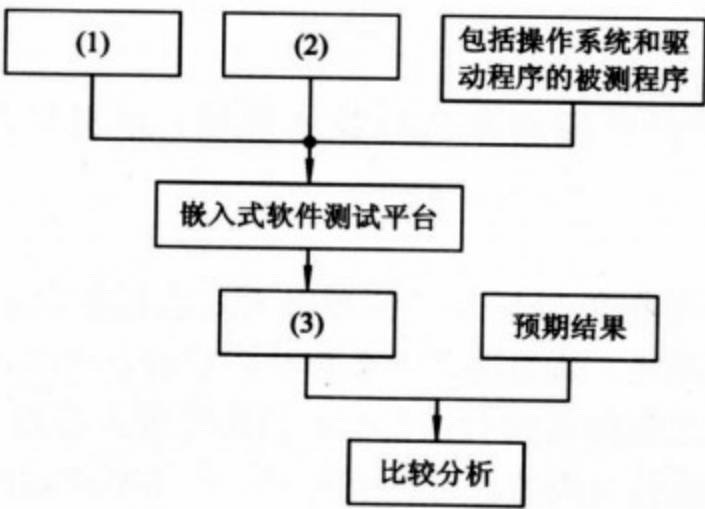


图 4-3 嵌入式软件的统一测试模型

【问题 2】(5 分)

从嵌入式系统的观点来看，白盒测试是很重要的一种测试方法，测试人员可以很清楚已经有多少代码被检查过了，而且通过测试结果，可以准确地预测出系统中的程序设计错误的数量规模。在白盒测试用例设计中，有语句覆盖、判定覆盖、条件覆盖、判定/条件覆盖、条件组合覆盖以及路径覆盖等几种常见的覆盖测试策略，其中(4)是最强的覆盖准则。

某嵌入式产品使用 C 语言书写的部分程序段如图 4-4 所示，其中 x, y 是两个已定义的整型变量。对该程序段进行覆盖测试时，必须适当地选取测试数据组。表 4-5 给出了可供选择的 4 组测试数据组，则实现语句覆盖至少应采用的测试数据组是(5)，实现条件覆盖至少应采用的测试数据组是(6)，实现路径覆盖至少应采用的测试数据组是(7)或(8)。

```

int a:=0;
if (x==0 && y>2)
    a:=1                /* A 语句 */
else
    { if (x<1 || y==1)
      else
        a:=2            /* B 语句 */
    }

```

图 4-4 某嵌入式产品使用 C 语言书写的部分程序段

表 4-5 测试数据组

变 量	x	y
测试数据组 I	0	3
测试数据组 II	1	2
测试数据组 III	-1	2
测试数据组 IV	3	1

【供选择的答案】

- (5)~(8) A. I 和 II 组 B. II 和 III 组 C. III 和 IV 组
 D. I 和 IV 组 E. I、II 和 III 组 F. II、III 和 IV 组
 G. I、III 和 IV 组 H. I、II 和 IV 组

【问题 3】(2 分)

程序的环路复杂度 $V(G)$ 也称为 McCabe 复杂性度量, 通常将它定义为控制流图的区域数, 它是构成基本路径集的独立路径数的上界, 可依此得出应该设计的测试用例数目。请计算图 4-4 中所给出的程序段的环路复杂度 $V(G)$ (独立线性路径数)。

【问题 4】(3 分)

要取得较好的覆盖测试效果, 需要借助一些具备特定功能特点的工具软件。图 4-5 示意了利用特定工具软件进行动态覆盖测试的基本流程。请结合图 4-2 给出的嵌入式软件覆盖测试的基本原理, 将图 4-5 中(9)~(11)空缺处填写完整。

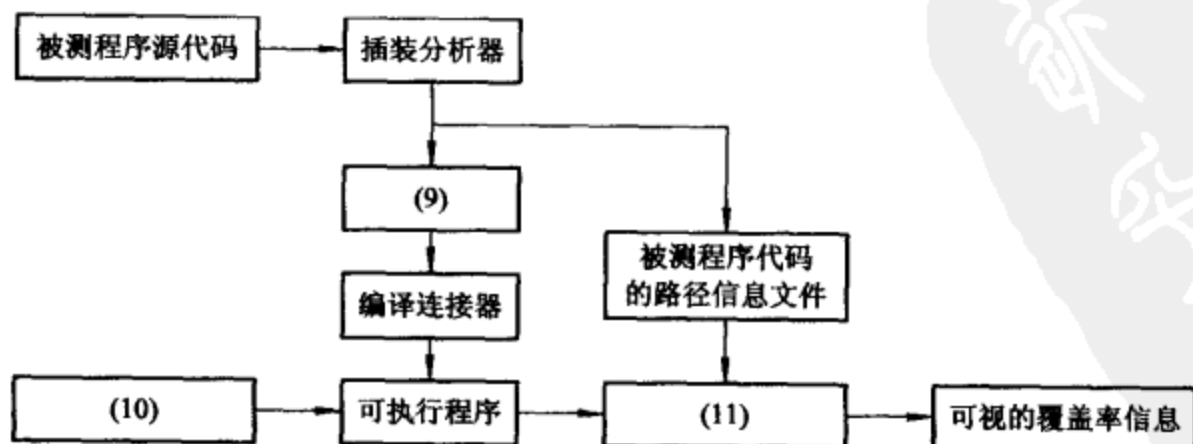


图 4-5 覆盖测试的基本流程图

【问题 5】(2 分)

结合嵌入式软件覆盖测试的基本原理图(图 4-2)和覆盖测试的基本流程图(图 4-5),说明在选购嵌入式软件覆盖测试工具时应重点关注哪两方面的选购因素。

二、要点解析**【问题 1】(3 分)**

这是一道要求读者掌握嵌入式软件规范化测试过程所包含的基本测试活动和嵌入式软件的统一测试模型的分析题。本题的分析思路如下:

(1) 进行嵌入式软件测试的根本目的是为了发现至今尚未发现的错误。一个规范化测试过程通常包含以下五方面的基本测试活动,分别是:① 拟定测试计划,② 编制测试大纲,③ 根据测试大纲设计和生成测试用例,④ 实施测试,⑤ 生成测试报告。

(2) 在第 1 步制定测试计划时,要根据测试任务充分考虑整个项目的开发时间和开发进度,以及一些人为因素和客观条件等,使得测试计划是可行的。测试计划的主要内容有测试的内容、进度安排、测试所需要的环境和条件、测试培训安排等。因此(1)空缺处可填入类似“测试任务”等内容。

(3) 第 2 步是制定一份完整的测试大纲,因为它是一个嵌入式软件测试活动的依据,大纲中应该明确、详尽地规定在测试中针对系统的每一项功能或特性所必须完成的基本测试项目和测试完成的标准。

(4) 第 3 步是根据测试大纲设计和生成测试用例。在设计测试用例的时候,不仅要设计有效合理的、正常的输入条件,也要包含不合理、失效的、异常的输入条件。在设计好测试用例后,要产生相应的测试设计说明文档,其内容主要有被测项目、输入数据、测试过程、预期输出结果等。可见,(2)空缺处可填入类似“测试用例”等内容。

(5) 第 4 步是实施测试。测试的实施阶段是由一系列的测试周期组成的。在每个测试周期中,测试人员和开发人员将依据预先编制好的测试大纲和准备好的测试用例,对被测软件或设备进行完整的测试。

(6) 第 5 步是生成测试报告。测试报告的主要内容有:① 对测试进行概要说明;② 列出测试的结论;③ 指出缺陷和错误;④ 提出一些建设性意见,如可采用的修改方法、各项修改预计的工作量以及修改的负责人等。因此(3)空缺处可填入类似“输出测试结果”或“生成测试报告”等内容。

【问题 2】(5 分)

这是一道要求读者掌握嵌入式软件测试过程中覆盖测试基本常识的分析题。本题的解答思路如下:

(1) 覆盖测试是一种白盒测试方法,通常应用在软件测试的早期,即单元测试阶段。其基本思路是以程序的内部结构为基础来设计测试用例,以覆盖尽可能多的程序内部逻辑结构,发现其中的错误和问题。测试人员必须事先拥有被测程序的规格说明书和程序清单。

(2) 在白盒测试用例设计中,有语句覆盖、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等几种常见的覆盖测试策略,这些策略的相关描述及发现错误的能力如表 4-6 所示。从表中可知,试题中所罗列的 6 种覆盖测试的策略中路径覆盖是最强的

覆盖准则, 因此(5)空缺处可填入“路径覆盖”。

表 4-6 几种常见的覆盖测试策略

发现错误能力	覆盖策略	描 述
弱 ↓	语句覆盖	每条语句都至少执行 1 次
	判定覆盖	每个判定的每个分支至少执行 1 次
	条件覆盖	每个判定的每个条件应取到各种可能的值
	判定/条件覆盖	同时满足判定覆盖和条件覆盖
强	条件组合覆盖	每个判定中各条件的每 1 种组合都至少出现 1 次
	路径覆盖	使程序中每一条可能的路径至少执行 1 次

(3) 画出图 4-4 所给程序段的程序流程图, 如图 4-6 所示。

(4) 结合表 4-6 中给出的语句覆盖测试策略的描述, 选择足够多的测试用例数据, 使被测程序中每条语句都至少执行 1 次。对于本案例的程序段, 可选择测试数据组 I ($x=0, y=3$), 满足“ $(x=0) \text{ and } (y>2)$ ”的判断条件, 使 A 语句被执行 1 次; 也可选择测试数据组 II ($x=1, y=2$), 不满足“ $(x<1) \text{ or } (y=1)$ ”的判断条件, 使 B 语句被执行 1 次。因此(5)空缺处应选择选项 A(I 和 II 组)。

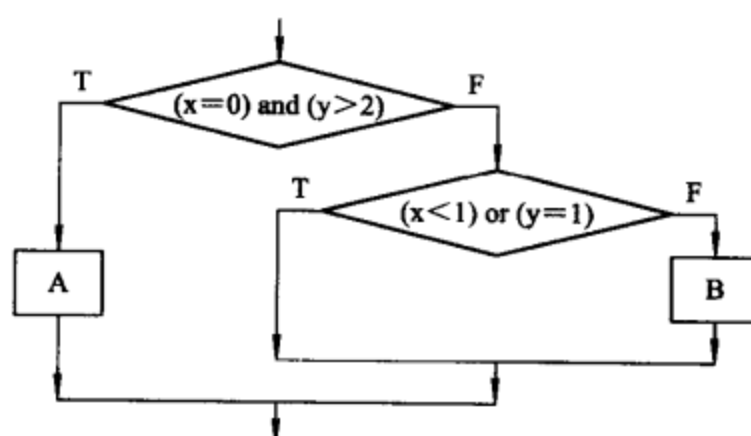


图 4-6 某嵌入式产品的部分程序流程图

(5) 结合表 4-6 中给出的判定覆盖测试策略的描述, 选择足够多的测试用例数据, 使得程序中每个判定都获得 1 次“真”值和“假”值, 或者说使程序中每 1 个取“真”分支和取“假”分支都至少通过 1 次。对于本案例的程序段, 可选择测试数据组 I ($x=0, y=3$), 使判断条件 $(x=0) \text{ and } (y>2)$ 的“真”分支执行 1 次; 也可选择测试数据组 II ($x=1, y=2$), 使判断条件 $(x<1) \text{ or } (y=1)$ 的“假”分支执行 1 次; 还可选择测试数据组 III ($x=-1, y=2$) 或选择测试数据组 IV ($x=3, y=1$), 使判断条件 $(x<1) \text{ or } (y=1)$ 的“真”分支执行 1 次。

(6) 条件覆盖测试策略的本质是构造一组测试用例数据, 使被测程序中每一个判定语句的每个逻辑条件的可能取值至少满足 1 次。对于本案例的程序段要实施条件覆盖测试, 可选择测试数据组 I ($x=0, y=3$), 以同时满足判断条件 $(x=0) \text{ and } (y>2)$ 中 $x=0$ 和 $y>2$ 的取值条件; 可选择测试数据组 III ($x=-1, y=2$), 满足判断条件 $(x<1) \text{ or } (y=1)$ 中 $x<1$ 的取值条件; 可选择测试数据组 IV ($x=3, y=1$), 满足判断条件 $(x<1) \text{ or } (y=1)$ 中 $y=1$ 的取值条件。因此(6)空缺处应选择选项 G(I、III 和 IV 组)。

(7) 结合表 4-6 中给出的判定/条件覆盖测试策略的描述, 构造一组测试用例数据, 使得判定中的每个逻辑条件取得各种可能的值, 并且使得每个判定取得各种可能的判定结果。结合以上(6)、(7)点的分析结论, 可知对于本案例的程序段, 选择测试数据组 I ($x=0, y=3$), 同时满足判断条件 $(x=0) \text{ and } (y>2)$ 中 $x=0$ 和 $y>2$ 的取值条件, 且使得该判断条件的“真”分支执行 1 次。

选择测试数据组 II ($x=1, y=2$), 使判断条件 $(x=0) \text{ and } (y>2)$ 的“假”分支执行 1 次; 同时不满足 $(x<1) \text{ or } (y=1)$ 的取值条件, 且使判断条件 $(x<1) \text{ or } (y=1)$ 的“假”分支执行 1 次。

选择测试数据组 III ($x=-1, y=2$), 满足判断条件 $(x<1) \text{ or } (y=1)$ 中 $x<1$ 的取值条件, 且使得使该判断条件的“真”分支执行 1 次, 使判断条件 $(x=0) \text{ and } (y>2)$ 的“假”分支执行 1 次。

也可选择测试数据组 IV ($x=3, y=1$), 满足判断条件 $(x<1) \text{ or } (y=1)$ 中 $y=1$ 的取值条件, 且使得使该判断条件的“真”分支执行 1 次, 使判断条件 $(x=0) \text{ and } (y>2)$ 的“假”分支执行 1 次。

(8) 对于条件组合覆盖测试策略, 其含义是选择足够多的测试用例数据, 使被测程序中每个判定中逻辑条件的各种组合都至少出现 1 次。对于本案例的程序段, 要实现条件组合覆盖测试工作所需的测试用例见表 4-7。

表 4-7 条件组合覆盖测试的测试用例

逻辑条件		测试数据组		逻辑条件		测试数据组	
$x=0$	$y>2$	x	y	$x<1$	$y=1$	x	y
F	F	1	1	F	F	1	2
F	T	1	3	F	T	3	1
T	F	0	1	T	F	-1	2
T	T	0	3	T	T	-1	1

(9) 结合表 4-6 中给出的路径覆盖测试策略的描述, 选择足够多的测试用例数据, 使被测程序中每一条可能的路径至少执行 1 次。本案例的程序段主要有 3 条路径: ① 执行判断条件 $(x=0) \text{ and } (y>2)$ “真”分支的路径; ② 执行判断条件 $(x=0) \text{ and } (y>2)$ “真”分支的路径; ③ 执行判断条件 $(x=0) \text{ and } (y>2)$ “假”分支的路径。

选择测试数据组 I ($x=0, y=3$), 同时满足判断条件 $(x=0) \text{ and } (y>2)$ 中 $x=0$ 和 $y>2$ 的取值条件, 使得该判断条件的“真”分支路径被执行 1 次。

选择测试数据组 II ($x=1, y=2$), 同时不满足判断条件 $(x<1) \text{ or } (y=1)$ 中 $x<1$ 和 $y=1$ 的取值条件, 使得该判断条件的“假”分支路径被执行 1 次。

选择测试数据组 III ($x=-1, y=2$), 满足判断条件 $(x<1) \text{ or } (y=1)$ 中 $x<1$ 的取值条件, 或者选择测试数据组 IV ($x=3, y=1$), 满足判断条件 $(x<1) \text{ or } (y=1)$ 中 $y=1$ 的取值条件, 使得使该判断条件的“真”分支路径被执行 1 次。

由以上分析可知, (7)、(8)空缺处可分别填入选项 E(I、II 和 III 组)、选项 H(I、II 和 IV 组)。

【问题3】(2分)

这是一道要求读者掌握嵌入式软件测试过程中计算程序环路复杂度的试题。本题的解答思路如下：

(1) 程序的环路复杂度 $V(G)$ 也称为 McCabe 复杂性度量，通常将它定义为程序控制流程图(如图4-6所示)的区域数，它是构成基本路径集的独立路径数的上界，可依此得出应该设计的测试用例数目。

(2) 在进行程序的基本路径测试时，从程序的环路复杂度可导出程序基本路径集中的独立路径条数，以确保程序中每个可执行语句至少执行一次所必须的测试用例数目的上界。

(3) 计算控制流图环路复杂性 $V(G)$ 的一种简单方法是： $V(G) = (\text{区域数}) = (\text{判断节点数}) + 1$ 。

(4) 阅读图4-6所示的程序控制流程图可知，该图的判断节点数为2个(($x=0$) and ($y>2$)和($x<1$) or ($y=1$))，所以图4-6中所给出的程序段的环路复杂度 $V(G) = 2 + 1 = 3$ 。顺便指出，该计算结果与本案例试题2的(9)点分析结果是一致的。

【问题4】(3分)

这是一道要求读者掌握利用工具软件进行嵌入式软件动态覆盖测试时所需的3个要素的分析题。本题的解答思路如下：

(1) 仔细阅读题干信息及图4-2，可知嵌入式软件覆盖测试的基本原理是：在目标机端，插装过的被测应用程序将覆盖信息发送到消息队列中，一个专用的任务负责在适当的时候将这些信息发送到宿主机端；宿主机端通过专门的覆盖信息接收模块接收相应的覆盖信息，并提交给分析工具分析和在线动态显示覆盖率的增长情况。

(2) 要取得较好的覆盖测试效果，需要借助一些可弥补人为测试的缺陷、具备特定功能特点的工具软件。这些工具软件一般具备以下几个功能特点：

① 分析软件内部结构，帮助制定覆盖策略及设计测试案例；

② 与适当的编译器结合，对被测软件实施自动插装，以便在其运行过程中生成覆盖信息并收集这些信息；

③ 根据搜集的覆盖信息计算覆盖率，帮助测试人员找到未被覆盖的软件部位，以改进测试案例，提高覆盖率。

(3) 由以上分析可知，在利用工具进行动态覆盖测试时需要3个要素：① 测试用例、② 插装后的被测代码、③ 收集覆盖信息并进行分析的工具本身。由图4-5所示的信息流程可知，代码插装由工具自动完成，通过执行测试用例，再由工具搜集覆盖信息并进行分析，就可以看到覆盖率信息指标。因此，图4-5中(9)空缺处可填入类似“插装后的被测代码”等内容；(10)空缺处可填入类似“测试用例”等内容；(11)空缺处可填入类似“覆盖信息收集及分析”等内容。

【问题5】(2分)

这是一道要求读者熟悉嵌入式软件覆盖测试工具选购要点的问答题。本题的分析思路如下：

(1) 由图4-2可知，① 在目标机方，应用任务与专门负责收集/上传覆盖信息的任务是通过消息队列来传递数据的，该消息队列可使用嵌入式操作系统的相应机制实现。② 这

个专门负责收集/上传覆盖信息的任务也可以被看做一个特殊的应用任务,也必须有嵌入式操作系统的支持,因为任务管理是嵌入式操作系统的基本功能之一。③ 目标机与宿主机之间的通信可以采用串行口或以太网网络接口等方式,对串行口的驱动或运行网络协议均需要使用嵌入式操作系统的相应程序组件。

由以上分析可知,在选购嵌入式软件覆盖测试的工具时,应关注支持嵌入式软件覆盖测试的工具与所使用的嵌入式操作系统的配合程度。

(2) 题干信息已指出,嵌入式应用程序的开发通常采用交叉开发方式,因此几乎所有的开发工具均要解决以下 3 个部分的问题:① 宿主机部分的功能;② 目标机部分的功能;③ 宿主机与目标机的连接问题。其中,宿主机与目标机的连接是个瓶颈,如果不同的工具要使用同一物理线路实现数据传输,则要解决对该物理线路(或者说硬件端口)的正确共享,即正确建立宿主机与目标机之间的物理/逻辑连接,解决数据信息的传输。因此,嵌入式软件覆盖测试的工具与其他嵌入式交叉开发工具的配合关系也是选购时重点考虑的因素之一。

三、参考答案

表 4-8 给出了本案例问题 1 至问题 5 的参考答案,供读者练习时参考,以便查漏补缺。读者也可依照所给出的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

表 4-8 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (3 分)	(1) 测试任务 (2) 测试用例 (3) 输出测试结果,或生成测试报告 (每空 1 分)	
【问题 2】 (5 分)	(4) 路径覆盖 (5) A 或(I 和 II 组) (6) G 或(I、III 和 IV 组) (7) E 或(I、II 和 III 组) (8) H 或(I、II 和 IV 组) (每空 1 分)	
【问题 3】 (2 分)	$V(G)=3$ (2 分)	
【问题 4】 (3 分)	(9) 插装后的被测代码 (10) 测试用例 (11) 覆盖信息收集及分析 (每空 1 分)	
【问题 5】 (2 分)	① 与所使用的嵌入式操作系统的配合程度 ② 与其他嵌入式交叉开发工具的配合关系(答案类似即可,每小点 1 分)	

4.2.3 案例 3 可移植性技术

一、案例描述

阅读以下关于可移植性技术的说明,结合网络拓扑图回答问题 1 至问题 5。(15 分)

【说明】

对于一个运行良好的嵌入式软件或其中的部分子程序，在今后的开发中可能会再次用在类似的应用领域。由于原有的代码已被反复应用、测试和维护，具有很好的稳定性，因此，在原有代码的基础上进行移植将会缩短开发周期，提高开发效率，降低开发成本。

嵌入式应用程序的移植，是指把应用程序从一个嵌入式操作系统平台移植到另一个操作系统平台。为了提高嵌入式应用程序的可移植性，在软件设计时通常应遵循如表4-9所示的一些开发原则。

表4-9 嵌入式应用程序开发的一些通用原则

原 则	描 述
① 在软件设计上，尽量采用层次化设计和模块化设计	在软件纵向结构上，下层为上层提供服务，上层去调用下层提供的服务。每一个层次都应该定义清晰的接口和功能，但分层的数量要合适。在整体软件的设计上，同一层软件的结构上要尽量形成模块化，且软件模块之间是相互独立的，一个模块的实现不依赖于其他模块的实现
② 在软件体系结构上，尽量在操作系统和应用程序之间引入一个虚拟机层(或称为操作系统抽象层)	把一些通用的、共性的操作系统 API 接口函数封装起来，在编写一个应用程序时，不是直接去调用实际操作系统的 API，而是使用虚拟层所提供的 API。在新的操作系统平台上仅要实现此虚拟层，而无需修改其他的代码。定义这个虚拟层时，要综合考虑现有的各种嵌入式操作系统的功能和特性，尽量采用标准的操作系统接口
③ 在功能服务的调用上，要尽量使用可移植的函数	如标准的 C 语言函数或自己编写的函数，尽量不要使用依赖于特定操作系统的 API 函数
④ 在数据类型上，用宏定义的方式定义一组可移植的数据类型，然后在应用程序的内部，只使用这些数据类型	这是由于 C 语言的数据类型与机器的字长和编译器有关 例如，可以用 INT32U 来表示无符号的 32 位整型数据，对于实际的编译器，可以定义为 #define INT32U unsigned int
⑤ 将不可移植的部分局域化	把不可移植的代码通过宏定义和函数的形式，分类集中于某几个特定的文件之中。在以后的移植过程中，既有利于迅速地对需要修改的代码进行定位，又可方便地进行修改
⑥ 提高代码的可重用性	例如，可以更好地抽象软件的函数，使之更加模块化，功能更专一，接口更简洁明了

【问题 1】(3 分)

移植嵌入式软件可分为无操作系统的软件移植和有操作系统的软件移植。对于把操作系统和应用软件作为一个整体，移植到一个新的嵌入式硬件平台上的情况，在系统移植时，真正需要移植的主要是(1)程序、(2)程序以及操作系统中同(3)密切相关的代码。

【问题 2】(3 分)

在嵌入式软件开发中，必须高度关注应用软件的可移植性和可重用性。为提高可移植性，Boot Loader 的实现一般分为 stage1 和 stage2 两大部分。依赖于 CPU 体系结构的代码，如设备初始化代码等，通常都放在(4)中，用(5)语言来实现；而 stage2 则采用 C 语言来实现。在移植时，主要的工作量在(6)的移植上，基本上要重新编写。

【问题 3】(3 分)

在 $\mu\text{C}/\text{OS}-\text{II}$ 实时操作系统中，为了方便移植， $\mu\text{C}/\text{OS}-\text{II}$ 的大部分代码都是用标准的 C 语言编写的。以下是 $\mu\text{C}/\text{OS}-\text{II}$ 操作系统部分文件。请指出在移植 $\mu\text{C}/\text{OS}-\text{II}$ 操作系统时，主要修改哪三个相关的文件？

INCLUDES. H	OS_MEM. C
OS_CFG. H	OS_CPU_C. C
OS_CPU. H	OS_Q. C
$\mu\text{C}/\text{OS}_\text{II}. \text{H}$	DOS_CORE. C
OS_CPU_A. ASM	OS_MBOX. C
$\mu\text{C}/\text{OS}_\text{II}. \text{C}$	OS_SEM. C
OS_TIME. C	OS_TASK. C

【问题 4】(3 分)

某嵌入式产品使用 C 语言书写的部分程序段(a)和(b)如图 4-7 所示，请指出哪个程序段属于可重入型代码？请在 200 字以内简要说明理由。

<pre>void swap_A(int *a, int *b) { int temp; temp = *a; *a = *b; *b = temp; }</pre> <p style="text-align: center;">(a)</p>	<pre>int temp; void swap_B(int *a, int *b) { temp = *a; *a = *b; *b = temp; }</pre> <p style="text-align: center;">(b)</p>
--	--

图 4-7 变量值交换函数

【问题 5】(3 分)

图 4-8 给出了通过一个简单的多任务程序来测试移植 $\mu\text{C}/\text{OS}-\text{II}$ 实时操作系统是否成功的部分程序代码。系统任务 SYS_Task 中 for 循环体的作用是什么？在测试程序语句中，时钟节拍中断语句是安排在多任务调度语句之前，还是安排在多任务调度语句之后？

```

OS_STK SYS_Task_Stack[STACKSIZE]={0, };
#define SYS_Task_Prio1
void SYS_Task(void * Id);
.....                //此处省略了其他任务的定义

int Main(int argc, char * * argv)
{  ARMTargetInit();      //do target (uHAL based ARM system) initialisation//
  OSInit();
  OSTaskCreate(SYS_Task, (void *)0, (OS_STK *) &SYS_Task_Stack[STACKSIZE-1],
SYS_Task_Prio);
  .....                //此处省略了其他任务的相关创建语句
  OSStart();              // start the game //
  return 0;
}

void SYS_Task(void * Id)
{  uHALr_InstallSystemTimer();    //打开 IRQ 和 TIMER1 中断, 启动系统时钟
  Uart_Printf("start system task\n");
  for(;;)
  {OSTimeDly(10000);
  }
}

```

图 4-8 移植 μ C/OS-II 实时操作系统的测试示例

二、要点解析

【问题 1】(3 分)

这是一道要求读者掌握嵌入式软件移植涉及哪些代码的基本常识题。本题的分析思路如下：

(1) 在移植嵌入式软件时可分为无操作系统的软件移植和有操作系统的软件移植。对于后者，又可以细分为两种情况，分别是把操作系统和应用软件作为一个整体进行移植和把应用软件移植到一个新的操作系统上。

(2) 把操作系统和应用软件作为一个整体，移植到一个新的嵌入式硬件平台上，真正需要移植的是与硬件直接打交道的部分，包括设备驱动层的软件和操作系统当中的部分代码，而其他的软件，如操作系统内核、中间件和应用软件，不用做任何修改。当然，在有些嵌入式操作系统中，如单体结构的操作系统，把设备驱动层的软件也集成在系统内核中，这时就要把相应的软件提取出来进行修改。

(3) 在系统移植时，真正需要移植的主要是引导加载程序 Boot Loader、设备驱动程序以及操作系统中同处理器密切相关的代码。

(4) Boot Loader 是在操作系统内核运行之前运行的一段小程序，其作用是初始化硬件设备和建立内存空间的映射图，以及将系统的软硬件环境带到一个合适的状态，以便为

最终调用操作系统内核准备好正确的环境。系统的 Boot Loader 程序通常安排在地址 0x00000000 处。

【问题 2】(3 分)

这是一道要求读者掌握嵌入式系统的 Boot Loader 技术的基本常识题。本题的解答思路如下：

(1) 在嵌入式软件开发中，必须高度关注应用软件的可移植性和可重用性。为提高可移植性，Boot Loader 的实现一般分为 stage1 和 stage2 两大部分。

(2) 从操作系统的角度看，Boot Loader 的总目标就是正确地调用内核来执行。其中，Boot Loader 的 stage1 实现依赖于 CPU 体系结构的代码，通常包括以下工作内容：

① 硬件设备初始化(屏蔽所有的中断、设置 CPU 的速度和时钟频率、RAM 初始化、关闭 CPU 内部指令/数据 Cache 等)；

② 为加载 stage2 准备 RAM 空间；

③ 拷贝 stage2 到 RAM 中；

④ 设置堆栈指针 SP；

⑤ 跳转到 stage2 的 C 语言入口点。

(3) stage2 可实现一些复杂的功能，通常包括以下工作内容：

① 初始化本阶段要使用到的硬件设备(如串行接口、计时器等)；

② 检测系统的内存映射；

③ 加载内核映像和根文件系统映像；

④ 设置内核的启动参数；

⑤ 调用内核。

(4) 可见，设备初始化代码等通常都放在 stage1 中，用汇编语言来实现；而 stage2 则采用 C 语言来实现。在移植时，主要的工作量在 stage1 的移植上，基本上要重新编写。

【问题 3】(3 分)

这是一道要求读者掌握移植 $\mu\text{C}/\text{OS}-\text{II}$ 操作系统时涉及哪些相关文件的基本常识题。本题的分析思路如下：

(1) 为了方便移植， $\mu\text{C}/\text{OS}-\text{II}$ 的大部分代码都是用标准的 C 语言编写的，不需要改动。只有少部分代码，尤其是那些跟 CPU 寄存器打交道的代码，需要针对具体的 CPU 类型进行修改，这些代码一般都是用汇编语言来写的。

(2) $\mu\text{C}/\text{OS}-\text{II}$ 操作系统的代码被分为三大部分：第一部分是与处理器无关的代码，如任务管理、任务调度、存储管理、信号量、邮箱、消息队列等；第二部分与系统的配置有关，应用程序开发人员可以通过修改这些配置文件来裁剪内核，选择自己需要的系统服务；第三部分是与处理器相关的代码，包括 OS_CPU.H、OS_CPU_A.ASM 和 OS_CPU_C.C 三个文件，在移植 $\mu\text{C}/\text{OS}-\text{II}$ 操作系统时，主要修改的就是这三个文件。

(3) OS_CPU.H 文件包括三部分的内容：① 1 个符号常量，用来设定处理器的栈的增长方向；② 3 个宏定义，用来关闭和打开中断；③ 10 个数据类型的定义，用来定义与编译器无关的数据类型(在操作系统内部，只使用这些数据类型，而不使用标准 C 的数据类型)。

(4) OS_CPU_A.ASM 文件是用汇编语言编写的 OS_CPU.ASM 文件中四个与处理器

相关的函数,包括任务切换和时钟中断服务程序等。

(5) OS_CPU_C.C 文件是用 C 语言编写的 10 个与操作系统相关的函数,通常只需要改写其中的一个函数,即任务堆栈的初始化函数。

【问题 4】(3 分)

这是一道要求读者掌握可重入函数代码概念的分析理解题。本题的解答思路如下:

(1) 可重入函数代码是指一段代码(如一个函数)可以被多个任务同时调用,而不必担心会破坏数据。换句话说,可重入函数代码在任何时候都可以被中断执行,不会因为在函数中断的时候被其他任务重新调用而影响函数中的数据。代码的可重入性是保证完成多任务的基础。

(2) 图 4-7 的(a)程序段使用局部变量 temp 作为 swap_A 函数的中间变量。由于 C 语言编译器通常把局部变量分配在栈空间中,因此多次调用同一个 swap_A 函数时,可以保证每次的 temp 变量值互不影响。由此可判断(a)程序段属于可重入型函数。

(3) 在图 4-7 的(b)程序段中,变量 temp 被定义成全局变量。由于 C 语言编译器通常把未带初始值的全局变量分配在 .bss 段中,当多次调用同一个 swap_B 函数时,temp 变量的值势必会受到相互影响。因此可判断(b)程序段属于非可重入型函数。

(4) 由以上分析可知,在 C 语言程序中,代码的可重入性除了使用局部变量之外,还需要 C 编译器的支持。另外本题也给出一些启发:① 尽量在函数中使用局部变量;② 调用某个函数之前关中断,调用后再开中断;③ 用信号量禁止该函数在使用过程中被再次调用等措施有助于使一个函数具有可重入性。

【问题 5】(3 分)

这是一道要求读者掌握实时操作系统移植测试工作内容的分析理解题。本题的解答思路如下:

(1) 从图 4-8 给出的测试代码可知,系统任务 SYS_Task 的创建工作涉及以下内容:

① 在程序开头定义了任务堆栈、任务优先级和任务函数声明;

② 在 main 函数中将任务函数名、任务堆栈和任务优先级三个参数传递给 OSTask Create()函数;

③ 编写具体的任务函数内容。

(2) 在系统任务 SYS_Task 函数中,首先调用 uHALr_InstallSystemTimer()函数打开系统 IRQ 和 TIMER1 中断,从而启动系统的时钟节拍;接着通过 Uart_Printf()函数向终端输出系统提示信息;然后通过 for 循环体将系统任务 SYS_Task 挂起一段时间,以使其他任务可以占用 CPU。

(3) 多任务操作系统的任务调度是基于时钟节拍中断的。在 $\mu\text{C}/\text{OS-II}$ 实时操作系统中,通过 TIMER1 定时器中断来产生节拍,借以实现时间的延时或期满等功能。

(4) 通常在测试程序设计时,必须在多任务调度语句之后安排允许时钟节拍中断的语句,即在 OSStart()函数调用之后再启动节拍中断。如果是在调用 OSStart()启动多任务调度之前就启动节拍中断, $\mu\text{C}/\text{OS-II}$ 实时操作系统将可能因运行状态不确定而崩溃。

三、参考答案

表 4-10 给出了本案例问题 1 至问题 5 的参考答案,供读者练习时参考,以便查漏补缺

缺。读者也可依照所给出的评分标准得出自己的测试分数，从而大致评估自己对这一方面知识点的掌握程度。

表 4-10 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (3 分)	(1) 引导加载程序 Boot Loader (2) 设备驱动程序 (3) 处理器 (每空 1 分)	
【问题 2】 (3 分)	(4) stage1 (5) 汇编 (6) stage1 (每空 1 分)	
【问题 3】 (3 分)	OS_CPU.H、OS_CPU_A.ASM、OS_CPU_C.C (每个文件 1 分)	
【问题 4】 (3 分)	(a) 程序段属于可重入型函数(1 分) (b) 程序段变量 temp 是局部变量，被分配在栈空间中，多次调用同一个函数时，可以保证每次的 temp 变量值互不影响(1 分) (c) 程序段变量 temp 是未带初始值的全局变量，被分配在 .bss 段中，多次调用同一个函数时，变量值相互影响(1 分)	
【问题 5】 (3 分)	for 循环体用于将系统任务 SYS_Task 挂起一段时间，以使其他任务可以占用 CPU(1 分) 必须在多任务调度语句之后安排允许时钟节拍中断的语句(1 分)，反之系统将可能因运行状态不确定而崩溃(1 分)	

4.2.4 案例 4 可靠性分析技术

一、案例描述

高可靠性和安全性是嵌入式系统的一大特点，也是嵌入式系统最重要和最突出的基本要求。可靠性是一个嵌入式系统能正常工作的能力，一般用平均故障间隔时间(MTBF)来度量。

阅读以下关于嵌入式软件可靠性分析方面的叙述，回答问题 1 至问题 4。(15 分)

【说明】

某嵌入式产品研发部门的谢工程师正在为某企业单位开发一个嵌入式实时应用软件——宽带路由器的 NanOs 程序，在编写软件维护与测试说明书时，涉及到如何定量地描述软件可靠性的问题。

谢工认为软件可靠性指的是在将要使用的指定环境下，软件能以用户可接受的方式正确运行任务所表现出来的能力。从定量角度看，似乎应当是该软件在约定的环境条件下和

在给定的时间区间内,按照软件规格说明的要求,成功地运行程序所规定功能的概率。但是,他感到要具体地做定量描述有些困难。

为此,谢工查阅到了本部门某个软件需求规格说明书中有关的一条说明,即错误(BUG)可采用蒙特卡罗(MonteCarlo)随机植入技术进行测试。采用错误随机植入模型估算出被测程序模块中存在的错误数具有以下几个优点:

- ① 工作方式相当直观,能在一定程度上反映出软件的质量;
- ② 虽然在技术上不完善,但至少产生了与软件质量相关的定量结果;
- ③ 在最坏的情况下,起码可用来衡量“测试工作的有效性”,在某种程度上作为测试是否能结束的一项标志。

谢工程师首先想到了曾经学到过采用蒙特卡罗随机统计技术确定不规则形状封闭图形面积的方法,即是采用一个大的矩形把待测的封闭图形完全包围在该大矩形的内部,由计算机大量生成在此矩形内均匀分布的“点”,然后计数清点一下在大矩形内总的“点”的个数和在封闭图形内的“点”的个数,可近似地得到以下计算表达式:

$$\text{封闭图形的面积} = \frac{\text{在封闭图形内点的个数} \times \text{已知的大矩形的面积}}{\text{大矩形内点的个数}}$$

于是谢工程师把这个思想应用于系统测试过程,先在某个长度(机器指令条数)为 10^5 的程序模块中随机地人为植入 10 个错误,然后由一个测试组进行测试,结果一共发现有 160 个错误,其中有 2 个是人为播种植入的错误。以上测试结果基于该测试过程中未引入新的错误。

【问题 1】(3 分)

请用蒙特卡罗错误随机植入模型估算出被测程序模块中将会遗留下多少个未被发现的隐藏错误。请简要列出计算式子及计算过程。

【问题 2】(5 分)

在进行上述分析后,谢工程师感到有些困惑,于是与本部门郭工程师进行了讨论。郭工程师告诉他可以借用硬件的 MTBF(平均无故障时间)或者 MTTF(失效的平均等待时间)作为软件可靠性的主要指标。谢工程师通过查阅相关资料得知,平均无故障时间 MTBF 与单位长度的剩余故障个数 ϵ_r 成反比,即

$$\text{MTBF} = \frac{1}{K \times \epsilon_r}$$

其中, K 的典型值取为 200。

同样谢工程师将这个思想应用于 NanOs 嵌入式系统测试过程。通过测试一段时间后,在 NanOs 程序中发现 795 个固有错误,且及时改正了测试中发现的错误。请估算出宽带路由器的 NanOs 程序的平均无故障时间。若要求把此平均无故障时间再提高 4 倍,则还需排除多少个错误?请简要列出计算式子及计算过程。

【问题 3】(4 分)

信息部门的吴总工程师向谢工程师建议了另一种测试方案作为“错误随机植入”测试方法的补充。即由 A 和 B 两组测试人员同时相互独立地测试同一份宽带路由器的 NanOs 程序某个模块的两个拷贝。假设经过了两周测试后, A 组发现的错误总数为 350 个, B 组发现的错误总数为 320 个,其中两个组发现的相同错误数目为 132 个。请估算在测试前此程

序原有多少个错误？并在 100 字以内简要说明使用这种估算方法所需假设的必要前提。请简要列出计算式子及计算过程。

【问题 4】(3 分)

假设该宽带路由器应用于某小型局域网，NanOs 程序去年全年处理业务约 250 万个单位数据，其中 5% 的业务处理中，每个业务需对宽带路由器嵌入式微处理器提交 7 次请求；其中 80% 的业务处理中，每个业务需对该嵌入式微处理器提交 5 次请求；其余 15% 的业务处理中，每个业务需对该嵌入式微处理器提交 3 次请求。根据以往统计结果，每年的业务增量为 12%，考虑到今后 3 年业务发展的需要，测试需按现有业务量的两倍进行。每年业务量集中在 8 个月，每个月 20 个工作日，每个工作日 9 小时。请按测试强度 80~20 原理估算该宽带路由器的嵌入式微处理器处理请求的能力，并简要列出计算过程。

二、要点解析

【问题 1】(3 分)

这是一道要求读者掌握定量分析嵌入式软件错误可靠性指标的计算题。本题的解答思路如下：

(1) 根据题干中给出的信息，可将错误随机植入模型定义成为估算某软件所含固有错误的具体数目 N ，可在此软件中人为随机植入 N_s 个错误数，然后进行比较软件的测试，如果通过测试发现了 M 个固有错误，则其中发现的人为随机植入错误数有 M_s 个。

(2) 由题干关键信息“测试过程中未引入新的错误”，可以进一步假设，人为随机植入 N_s 个错误数后在故障的出现概率等同于植入错误数之前故障的出现概率，即将程序模块单位长度故障率认为是一个常数。

(3) 借鉴题干中给出的“封闭图形的面积 = $\frac{\text{在封闭图形内点的个数} \times \text{已知的大矩形的面积}}{\text{大矩形内点的个数}}$ ”

计算式子的思路，得到 $\frac{N}{M} = \frac{N_s}{M_s}$ ，该等式变形后得到 $N = \frac{M}{M_s} \times N_s$ ，即将“软件固有错误数 N ”看成是“封闭图形的面积”；将“人为植入错误数 N_s ”看成是“已知的大矩形的面积”；将“测试中发现的固有错误数 M ”、“测试中发现的人为植入错误数 M_s ”分别看成是“在封闭图形内点的个数”、“大矩形内点的个数”。

(3) 将题干中给出的数据整理后得出， $N_s = 10$ ， $M = 160$ ， $M_s = 2$ 。于是可估算出程序中错误的总数 N ：

$$N = \frac{M}{M_s} \times N_s = \frac{160 \times 10}{2} = 800 \text{ 个}$$

(4) 被测程序模块中将会遗留下未被发现的隐藏错误数 $= 800 - 160 - (10 - 2) = 640 - 8 = 632$ 个。

(5) 另外说明一点，蒙特卡罗错误随机植入模型存在以下几个局限性：① 由于所有错误不可能等概率出现，而且错误有着连带相关性（一个错误可能隐藏另一错误），因此要想使随机植入的错误有助于正确地推算出固有的错误数时，如何有效地在程度模块中选择和植入这类错误相对很困难；② 在检测错误时，错误一般不会等概率地被发现，而且在修复错误时也经常会引出一些新的错误，可见很难用简单的公式获得很理想的估计值；③ 随机植入的错误本身会增加检测发现错误和修改错误的工作量。

【问题 2】(5 分)

这是一道要求读者掌握定量计算嵌入式软件平均无故障时间(MTBF)指标的综合题。本题的分析思路如下:

(1) 设已知的被测程序长度为 L , 已检测到且修正过的错误数为 E_c , 依题意得 $L=10^5$, $E_c=795$ 。

(2) 根据 Shooman 公式—— $\epsilon_r = \frac{N-E_c}{L}$, 以及题干已给出的计算公式 $MTBF = \frac{1}{K \times \epsilon_r}$, 得出宽带路由器 NanOs 程序的平均无故障时间 MTBF 为

$$MTBF = \frac{1}{K \times \epsilon_r} = \frac{L}{K \times (N - E_c)} = \frac{10^5}{200 \times (800 - 795)} = 100 \text{ 小时}$$

(3) 试题要求将此平均无故障时间再提高 4 倍, 即 $MTBF = 500$ 小时, 由 $MTBF = \frac{1}{K \times \epsilon_r} = \frac{L}{K \times (N - E_c)}$ 得

$$E_c = N - \frac{L}{K \times MTBF} = 800 - \frac{10^5}{200 \times 500} = 800 - 1 = 799 \text{ 个}$$

(4) 由于已检测到且修正过的错误数有 795 个, 因此再需要改正的错误数个数是: $799 - 795 = 4$ 个。

(5) 通过本题的计算过程可以看出, MTBF 是用户可预测性和软件中存在有的各类错误的一个复杂的函数。即使两个软件用来提供同样的功能并有着相同的错误数目, 在不同的用户使用情况下也会有不同的 MTBF (与用户的可预测性有关); 功能上大体相同的两个软件在相同用户使用条件下, 由于软件有不同的错误数, 因此也会得到不同的 MTBF 值 (此时错误数起主要作用)。

【问题 3】(4 分)

这是一道要求读者运用 Hyman 分别测试法估算软件代码错误总数的计算分析题, 本题的主要解答思路如下:

(1) 仔细阅读试题信息, 将 A、B 两个测试小组发现的错误数分别看成是两个“独立”事件, 设 N 为宽带路由器 NanOs 程序中被测程序模块存在的故障总数, N_A 为 A 组检测到的错误数, $P(A)$ 为 A 组检测到的错误数在被测程序模块故障总数中发生的概率, N_B 为 B 组检测到的错误数, $P(B)$ 为 B 组检测到的错误数在被测程序模块故障总数中发生的概率, N_{AB} 为 A、B 两组发现的相同错误数, $P(A \cap B)$ 为 A、B 两组发现的相同错误数发生的概率。

(2) 根据事件 A、B 相互独立的充分必要条件 $P(A \cap B) = P(A)P(B)$, 即独立事件的 Bayes 概率计算公式, 得

$$P(A) = \frac{N_A}{N}, P(B) = \frac{N_B}{N}, P(A \cap B) = \frac{N_{AB}}{N}$$

$$\frac{N_{AB}}{N} = \frac{N_A}{N} \times \frac{N_B}{N}$$

即

$$N = \frac{N_A \times N_B}{N_{AB}}$$

(3) 由试题给出的数据知, $N_A = 350$, $N_B = 320$, $N_{AB} = 132$ 。利用计算公式 $N = \frac{N_A \times N_B}{N_{AB}}$ 估算出测试前此程序原有的错误数, $N = \frac{N_A \times N_B}{N_{AB}} = \frac{350 \times 320}{132} \approx 848.485$ 个, 取整数 849 个。

(4) 独立测试方案的估算前提是: ①两组发现的不同错误数所占的比例相对是很低的; ②两周来发现的错误在全部错误中有着代表性。

(5) 顺便指出, 由于目前所进行的仅仅是模块测试, 即部件级测试, 还没有考虑到集成测试和系统测试中将必然会出现的 NanOs 程序错误数量, 因此可以预知总的 NanOs 程序错误数必然是大于 849 个。

【问题 4】(3 分)

这是一道要求读者根据实际应用环境计算嵌入式微处理器处理请求能力的综合计算题, 本题的解答思路如下:

(1) 该宽带路由器的嵌入式微处理器处理请求的 80~20 原理, 是指每个工作日中 80% 的业务在 20% 的时间内完成。

(2) 每年总的请求数 M_1 为

$$\begin{aligned} M_1 &= (250 \times 5\% \times 7 + 250 \times 80\% \times 5 + 250 \times 15\% \times 3) \times 2 \text{ 万次/年} \\ &= (87.5 + 1000 + 112.5) \text{ 万次/年} \\ &= 1200 \text{ 万次/年} \end{aligned}$$

注意, 由于试题已指出“每年的业务增量为 12%, 考虑到今后 3 年业务发展的需要, 测试需按现有业务量的两倍进行”, 其中“12%”是个无关信息量, 计算每年总的请求数时需乘以 2(两倍)。

(3) 根据“每年业务量集中在 8 个月, 每个月 20 个工作日”可计算出每天请求数 M_2 。

$$M_2 = \frac{1200 \text{ 万次/年}}{8 \times 20 \text{ 天}} = 7.5 \text{ 万次/天}$$

(4) 根据 80~20 原理和试题中“每个工作日 9 小时”可计算出每秒请求数 M_3 。

$$M_3 = \frac{75\,000 \text{ 次/天} \times 80\%}{9 \times 3600 \times 20\% \text{ 秒}} = 9.259 \text{ 次/秒}$$

(5) 通过以上分析, 可以得出该宽带路由器的嵌入式微处理器处理请求的能力应达到 10 次/秒。

(6) 顺便指出, 本题涉及的内容属于负载压力测试中的疲劳强度测试。负载压力测试是嵌入式系统的性能测试的一项重要工作, 它是指在一定约束条件下测试系统所能承受的并发用户量、运行时间、数据量, 以确定系统所能承受的最大负载压力。它有助于确认被测系统是否能够支持性能需求, 以及预期的负载增长等。负载压力测试不只是关注不同负载场景下的响应时间等指标, 它也要通过测试来发现在不同负载场景下会出现的诸如速度变慢等问题的原因。

三、参考答案

表 4-11 给出了本案例问题 1 至问题 4 的参考答案, 供读者练习时参考, 以便查漏补缺。读者也可依照所给出的评分标准得出自己的测试分数, 从而大致评估自己对这一方面知识点的掌握程度。

表 4-11 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (3 分)	<p>设软件固有错误数为 N，人为植入错误数为 N_s，测试中发现的固有错误数为 M，测试中发现的人为植入错误数为 M_s，$N = \frac{M}{M_s} \times N_s = \frac{160 \times 10}{2} = 800$ 个 (2 分)</p> <p>被测程序模块中将会遗留下未被发现的隐藏错误数 $= 800 - 160 - (10 - 2) = 640 - 8 = 632$ 个 (1 分)</p>	
【问题 2】 (5 分)	<p>设已知的被测程序长度为 L，已检测到且修正过的错误数为 E_c</p> $MTBF = \frac{1}{K \times \epsilon_r} = \frac{L}{K \times (N - E_c)} = \frac{10^5}{200 \times (800 - 795)} = 100 \text{ 小时} \quad (2 \text{ 分})$ $E_c = N - \frac{L}{K \times MTBF} = 800 - \frac{10^5}{200 \times 500} = 800 - 1 = 799 \text{ 个} \quad (2 \text{ 分})$ <p>需要改正的错误数是 $799 - 795 = 4$ 个 (1 分)</p>	
【问题 3】 (4 分)	<p>设 N 为宽带路由器 NanOs 程序中被测程序模块存在的故障总数，N_A 为 A 组检测到的错误数，N_B 为 B 组检测到的错误数，N_{AB} 为 A、B 两组发现的相同错误数</p> $N = \frac{N_A \times N_B}{N_{AB}} = \frac{350 \times 320}{132} \approx 848.485 \text{ 个，取整数 } 849 \text{ 个} \quad (2 \text{ 分})$ <p>独立测试方案的估算前提是：① 两组发现的不同错误数所占的比例相对是很低的 (1 分)；② 两周来发现的错误在全部错误中有着代表性 (1 分)</p>	
【问题 4】 (3 分)	<p>① 每年总的请求数</p> $M_1 = (250 \times 5\% \times 7 + 250 \times 80\% \times 5 + 250 \times 15\% \times 3) \times 2 \text{ 万次/年}$ $= (87.5 + 1000 + 112.5) \text{ 万次/年} = 1200 \text{ 万次/年} \quad (1 \text{ 分})$ <p>② 每天请求数 $M_2 = (1200 \text{ 万次/年}) / (8 \times 20 \text{ 天}) = 7.5 \text{ 万次/天} \quad (1 \text{ 分})$</p> <p>③ 每秒请求数 $M_3 = (75 \text{ 000 次/天} \times 80\%) / (9 \times 3600 \times 20\% \text{ 秒}) = 9.259 \text{ 次/秒}$</p> <p>④ 该宽带路由器的嵌入式微处理器处理请求的能力应达到 10 次/秒 (1 分)</p>	

4.2.5 案例 5 性能评价技术

一、案例描述

阅读以下关于嵌入式系统的性能评价的技术说明，回答问题 1 至问题 3。(15 分)

【说明】

嵌入式系统的性能评价是指为了一定的目的，按照一定的步骤，选用一定的度量项目，通过建模和试验，对嵌入式系统的性能进行测试并对测试结果作出解释的技术。性能评估定量计算通常采用一个 M/M/1 排队模型。

某嵌入式实时系统的性能评估模型可等效为图 4-9 所示，即该系统主要由 3 个子系

统串联连接。用 ρ_1 、 ρ_2 、 ρ_3 分别表示子系统 1、子系统 2 和子系统 3 的资源利用率。在 M/M/1 排队模型中资源利用率的计算公式是

$$\text{资源利用率 } \rho = \frac{\lambda}{u} = \frac{\text{平均到达事务数}}{\text{平均处理事务数}}$$

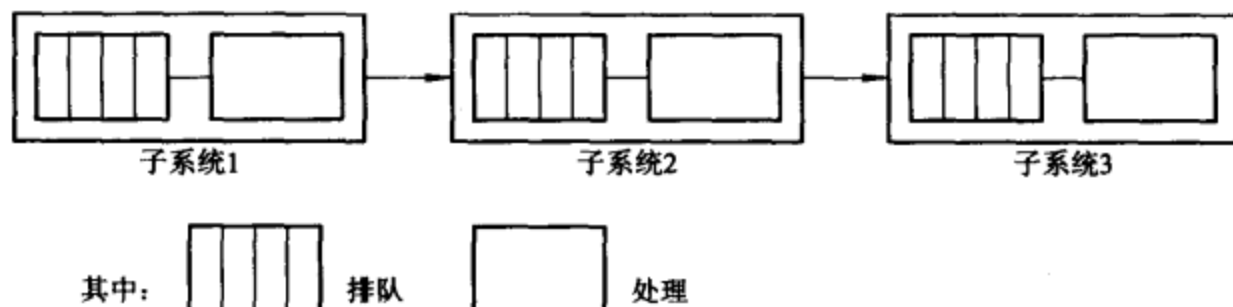


图 4-9 新 OA 系统的 M/M/1 排队模型

现已测得子系统 1、子系统 2 和子系统 3 对单个任务请求的平均处理时间分别为 30 ms、40 ms 和 10 ms。根据历史经验，每秒平均有 20 个任务，且任务到达间隔服从负指数分布。

基于以上测量值及经验值，可用排队论估算当并发到达的任务数量增加时的响应时间，即系统的实时性。每个队列的平均系统时间定义为平均等待时间与平均处理时间之和（单位：秒）。可用以下公式计算：

$$\text{平均系统时间} = \frac{\text{平均处理时间}}{1 - \text{利用率 } \rho}$$

【问题 1】(5 分)

由于嵌入式系统的复杂性，因此其性能难以用几个简单的技术指标来定量描述。那么对于嵌入式系统性能可从哪几个方面进行定性度量？请在 200 字以内简要说明理由。

【问题 2】(6 分)

评价嵌入式系统性价比的关键参数之一是资源利用率，它用于度量资源忙的时间所占的百分比。请计算本嵌入式实时系统的以下参数（计算结果在小数点后保留 1 位）。

- ① 子系统 1 的资源利用率 $\rho_1 = ()$ ；
- ② 子系统 2 的资源利用率 $\rho_2 = ()$ ；
- ③ 子系统 3 的资源利用率 $\rho_3 = ()$ ；
- ④ 整个嵌入式系统的平均响应时间 R 为 $()$ ms。

【问题 3】(4 分)

假设在峰值时间点，本嵌入式实时系统并发处理应用任务 1 的请求有 10 个，处理应用任务 2 的请求有 15 个，处理应用任务 3 的请求有 5 个，表 4-12 是对这些应用任务的相关描述。请计算本嵌入式实时系统传输数据的峰值流量（请写出具体的计算过程）。

表 4-12 参数描述

任务名称	微处理器处理时间	数 据 量	数据传输时间
应用任务 1	300 ms	由 2500 个记录构成，每个记录 51.2 B	10 s
应用任务 2	185 ms	由 1000 个记录构成，每个记录 96 B	15 s
应用任务 3	25 ms	由 200 个记录构成，每个记录 18 B	4 s

二、要点解析

【问题 1】(5 分):

嵌入式系统性能可从以下 10 个方面进行度量:

① 部件的性能指标和综合性指标。部件的性能指标如字长、存储容量、存取时间、加法时间以及编译速度等。吞吐率、实时性和各种利用率这三类综合性指标都是量化的指标,也是性能评价重点研究的对象。

② 可靠性。高可靠性和安全性是嵌入式系统的一大特点。其中,可靠性是嵌入式系统最重要、最突出的基本要求,用来衡量一个嵌入式系统能持续正常工作的能力,一般用平均故障间隔时间(MTBF)来度量。

③ 可维护性。它是指系统失效后在固定时间内可修复到规定功能的能力,一般用平均修复时间 MTTR 来表示。

④ 可用性。它表示需要使用嵌入式系统时,系统所能提供使用的程度,人机接口界面的友好和易用程度以及系统的稳定程度等,有时也指系统实际可用时间与计划提供使用时间的比值。

⑤ 系统的功耗,即系统电能消耗。嵌入式系统对功耗具有严格的要求,从微处理器内部的电源模式管理到外围接口电路,都要求尽量降低系统的功耗。

⑥ 环境适应性。由于大多数嵌入式系统所处的工作环境较恶劣,因此嵌入式系统必须适应用户环境要求,这样才能保证系统长期、稳定地工作。

⑦ 通用性。主要指某一个嵌入式系统设计投入使用后,以后设计类似系统时,能够在前一个系统的基础上添加或减少某些功能部件从而构成一个新的系统。

⑧ 安全性。它是指嵌入式系统中的程序和数据等信息的安全程度,例如数据库中的数据不被破坏或不被非法修改等。

⑨ 保密性。它是指确保系统内信息不被未授权人员非法存取等。

⑩ 可扩展性。它是指嵌入式系统的软、硬件扩充能力,以提高系统性能。例如预留扩展插槽,以方便以后可增加插件板连到原系统中。

【问题 2】(6 分)

这是一道要求读者掌握嵌入式实时系统资源利用率计算方法的综合题。本题的计算思路如下:

① 单位时间内资源繁忙的时间所占的百分比通常用资源利用率度量,它是评价嵌入式系统性价比的关键参数。通常有总利用率和净利用率两类,用户数据处理和开销一起占总容量的百分比称为总利用率;不考虑系统开销,用户数据处理占总容量的百分比称为净利用率,也称有效利用率。本试题是关于净利用率的计算问题。

② 根据题目所给的关键信息“子系统 1、子系统 2 和子系统 3 对单个任务请求的平均处理时间分别为 30 ms、40 ms 和 10 ms”可计算出,子系统 1 每秒平均处理任务数 $N_1 = \frac{1\text{ s}}{30\text{ ms}} = \frac{100}{3}$ 个,同理子系统 2 和子系统 3 每秒平均处理任务数分别为 $N_2 = \frac{1\text{ s}}{40\text{ ms}} = \frac{100}{4} = 25$ 个、 $N_3 = \frac{1\text{ s}}{10\text{ ms}} = \frac{100}{1} = 100$ 个。

③ 再依据题目给出的“每秒平均有 20 个任务到达”和“利用率 $\rho = \frac{\lambda}{u} = \frac{\text{平均到达事务数}}{\text{平均处理事务数}}$ ”这两个关键信息进行计算:

$$\text{子系统 1 的利用率 } \rho_1 = \frac{20}{100/3} = 60\%;$$

$$\text{子系统 2 的利用率 } \rho_2 = \frac{20}{25} = 80\%;$$

$$\text{子系统 3 的利用率 } \rho_3 = \frac{20}{100} = 20\%;$$

④ 根据题目中给出的计算公式“平均系统时间 = 平均处理时间 / (1 - 利用率 ρ)”可知, 整个嵌入式系统的平均响应时间

$$\begin{aligned} R &= \frac{\text{子系统 1 的平均处理时间}}{1 - \rho_1} + \frac{\text{子系统 2 的平均处理时间}}{1 - \rho_2} + \frac{\text{子系统 3 的平均处理时间}}{1 - \rho_3} \\ &= \frac{30 \text{ ms}}{1 - 60\%} + \frac{40 \text{ ms}}{1 - 80\%} + \frac{10 \text{ ms}}{1 - 20\%} \\ &= 287.5 \text{ ms}. \end{aligned}$$

【问题 3】(4 分)

这是一道要求读者熟练应用数据传输速率公式计算具体应用的综合题。本题的计算思路如下:

① 由于在峰值时间, 本嵌入式实时系统并发处理应用任务 1 的请求有 10 个, 而应用任务 1 由 2500 个记录构成, 每个记录 51.2 B, 因此峰值时间段本嵌入式实时系统并发处理应用任务 1 的数据量

$$D_1 = 2500 \times 51.2 \times 8 \text{ bit} \times 10 \text{ 个} = 10\,240\,000 \text{ bit} = 1.024 \times 10^7 \text{ bit}$$

注意, 计算时要把 51.2 B 转换成 $51.2 \times 8 \text{ bit}$, 以便统一单位, 为后面计算数据传输速率奠定基础。

② 同理, 本嵌入式实时系统并发处理 15 个应用任务 2 的请求, 而应用任务 2 由 1000 个记录构成, 每个记录 96 B, 因此峰值时间段本嵌入式实时系统并发处理应用任务 2 的数据量

$$D_2 = 1000 \times 96 \times 8 \text{ bit} \times 15 \text{ 个} = 1.152 \times 10^7 \text{ bit}$$

③ 本嵌入式实时系统并发处理 5 个应用任务 3 的请求, 而应用任务 3 由 200 个记录构成, 每个记录 18 B, 因此峰值时间段本嵌入式实时系统并发处理应用任务 3 的数据量

$$D_3 = 200 \times 18 \times 8 \text{ bit} \times 5 \text{ 个} = 144\,000 \text{ bit} = 0.144 \times 10^7 \text{ bit}$$

④ 本嵌入式实时系统传输数据的峰值流量, 是指这 30 个 (即 $10 + 15 + 5$) 并发应用任务请求同时与本嵌入式实时系统进行相互通信的数据传输速率。设该峰值流量为 S , 根据数据传输速率计算公式 $S = D/t$ 可得

$$\begin{aligned} S &= \frac{D}{t} = \frac{D_1}{t_1} + \frac{D_2}{t_2} + \frac{D_3}{t_3} = \frac{1.024 \times 10^7 \text{ bit}}{10 \text{ s}} + \frac{1.152 \times 10^7 \text{ bit}}{15 \text{ s}} + \frac{0.144 \times 10^7 \text{ bit}}{4 \text{ s}} \\ &= (1.024 + 0.768 + 0.036) \times 10^6 \text{ b/s} = 1.828 \times 10^6 \text{ b/s} = 1.828 \text{ Mb/s} \end{aligned}$$

注意, 常见的一种错误计算过程是

$$S = \frac{D}{t} = \frac{D_1 + D_2 + D_3}{t_1 + t_2 + t_3} = \frac{1.024 \times 10^7 \text{ bit} + 1.152 \times 10^7 \text{ bit} + 0.144 \times 10^7 \text{ bit}}{10 \text{ s} + 15 \text{ s} + 4 \text{ s}} \approx 0.8 \text{ Mb/s}$$

产生这种错误计算结果原因有二：一是含糊地理解数据传输速率计算公式所表示的意义；二是没注意到题目中关键信息“嵌入式实时系统并发处理应用任务”，其中“并发处理”可理解为每个应用任务请求是独立到达本嵌入式实时系统的。

三、参考答案

表4-13给出了本案例问题1至问题3的参考答案，供读者练习时参考，以便查漏补缺。读者也可依照所给出的评分标准得出自己的测试分数，从而大致评估自己对这一方面知识点的掌握程度。

表4-13 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题1】 (5分)	① 部件的性能指标和综合性指标；② 可靠性；③ 可维护性；④ 可用性；⑤ 系统的功耗；⑥ 环境适应性；⑦ 通用性；⑧ 安全性；⑨ 保密性；⑩ 可扩展性(每小点0.5分)	
【问题2】 (6分)	(1) 子系统1的利用率 $\rho_1 = \frac{2}{100/3} = 60\%$ (1分) (2) 子系统2的利用率 $\rho_2 = \frac{20}{25} = 80\%$ (1分) (3) 子系统3的利用率 $\rho_3 = \frac{20}{100} = 20\%$ (1分) (4) 整个嵌入式系统的平均响应时间 $R = 287.5 \text{ ms}$ (3分)	
【问题3】 (4分)	本嵌入式实时系统传输数据的峰值流量为 1.828 Mb/s，详细计算过程参见要点解析(4分)	

第5章 嵌入式Linux系统技术

5.1 学习要点

5.1.1 考纲要求

根据考试大纲中相应的考核要求，在Linux系统的服务应用技术方面，要求考生主要掌握以下方面的内容：

一、嵌入式操作系统应用技术

1. 时间管理
 - 系统时间；
 - 时钟中断。
2. 内存管理
 - 静态内存管理；
 - 动态内存管理。
3. 任务管理和任务间的通信
 - 任务间的通信机制；
 - 信号量；
 - 邮箱；
 - 消息队列。
4. 异常处理
 - 异常处理方法；
 - 中断优先级处理方法；
 - 系统调用。
5. 嵌入式文件系统应用技术
6. 嵌入式系统图形用户接口(GUI)应用技术
7. 嵌入式系统数据库应用技术

二、系统级软件设计技术

1. 嵌入式系统固件与系统初始化设计
2. 设备驱动程序设计
3. 硬件抽象层、板级支持包设计
4. 嵌入式软件的移植技术



5.1.2 学习建议

嵌入式 Linux 系统属于开放源代码单体结构的嵌入式操作系统，具有免费、开源、高性能和可裁剪的内核、丰富的硬件和软件资源支持、较完善的网络通信和文件管理机制以及优良的性能等特点，适合于网络设备、移动通信、数字家电和工业控制等领域的应用。

同样在应试环境中，可以预见嵌入式 Linux 系统技术方面的知识点可能成为今后下午试题考核的重点内容，且可能占较大的分值比重。因此对一些经典的知识点要多花力气、多总结、多比较、找异同点、抓规律，多阅读相关材料，多动笔做针对性的练习，并从试题中多思考其出题的风格、出发点、形式等，以便考试时能灵活变通，节约解题思考时间。

这一章节考试难度要求较高，所考查的知识点可能比较细腻，这就对考生在掌握知识的深度上提出了较高的要求，大致达到“深入掌握”的层次。在解答这类试题的过程中，考生除了熟悉有关理论、方法和配置步骤之外，还需尽可能地创造条件安装一台 Linux 系统的实验机，并熟悉其常用的操作命令和软件，读懂相关的配置文件信息。按照考试可能具有的命题规律，笔者建议考生要紧紧围绕新考试大纲中相应的考核要求来准备，并应在 Linux 系统的命令操作与应用软件配置等知识点上多下功夫，尽量从本书所罗列的配置案例中获取经验，以便考试时能灵活发挥。本章也将力求以发展的眼光，从实用的角度来预测和挖掘与嵌入式 Linux 系统相关的考核点，以增强考生学习相关知识点的目的性。

通过本章学习，可获得以下理论知识点及工程实践经验：

- ① inittab 配置文件；
- ② Samba 应用配置文件；
- ③ RPM 的操作命令；
- ④ TCP/IP 网络配置命令。

5.2 典型案例

5.2.1 案例 1 inittab 配置文件

一、案例描述

阅读以下 Linux 实时操作系统中 inittab 配置文件的技术说明，根据试题要求回答问题 1 至问题 7。(15 分)^①

【说明】

Linux 实时操作系统的安装需经过“安装前的准备”、“启动安装程序”、“安装”等几个阶段。其中，“安装前的准备”工作包括硬盘分区、制作安装引导盘等。“启动安装程序”阶段需要进行安装程序的引导，根据所选择安装介质的不同，可以使用光盘或软盘两种引导方式。“安装”阶段基于图形用户安装界面，包括配置引导、复制文件及创建引导盘等过程。

^① 本试题的题干描述参考了 2006 上半年网络工程师下午试题 2 的相关内容。

Linux 实时操作系统开机引导时首先启动内核，由内核检查和初始化硬件设备，载入设备的驱动程序模块，安装 root 文件系统，然后内核将启动一个名为 init 的进程。在 init 运行完成并启动其他必要的后续进程后，系统开始运行，引导过程结束。init 进程启动时需要读取 inittab 配置文件，该文件确定 init 在系统启动和关机时的工作特性。

典型的 inittab 文件内容见以下清单：

```
#
# inittab This file describes how the INIT process should set up
#   the system in a certain run-level.
#
# Default runlevel. The runlevels used by RHS are:
# 0 - halt (Do NOT set initdefault to this)
# 1 - Single user mode
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
# 3 - Full multiuser mode
# 4 - unused
# 5 - X11
# 6 - reboot (Do NOT set initdefault to this)
#
id: 5: initdefault:
# System initialization.
si: : sysinit: /etc/rc.d/rc.sysinit
10: 0: wait: /etc/rc.d/rc 0
11: 1: wait: /etc/rc.d/rc 1
12: 2: wait: /etc/rc.d/rc 2
13: 3: wait: /etc/rc.d/rc 3
14: 4: wait: /etc/rc.d/rc 4
15: 5: wait: /etc/rc.d/rc 5
16: 6: wait: /etc/rc.d/rc 6
# Trap CTRL-ALT-DELETE
ca: : ctrlaltdel: /sbin/shutdown -t3 -r now
# When our UPS tells us power has failed, assume we have a few minutes
# of power left. Schedule a shutdown for 2 minutes from now.
# This does, of course, assume you have powerd installed and your
# UPS connected and working correctly.
pf: : powerfail: /sbin/shutdown -f -h +2 "Power Failure; System Shutting Down"
# If power was restored before the shutdown kicked in, cancel it.
pr: 12345: powerokwait: /sbin/shutdown -c "Power Restored; Shutdown Cancelled"
# Run gettys in standard runlevels
1: 2345: respawn: /sbin/mingetty tty1
2: 2345: respawn: /sbin/mingetty tty2
3: 2345: respawn: /sbin/mingetty tty3
4: 2345: respawn: /sbin/mingetty tty4
```

```
5: 2345: respawn: /sbin/mingetty tty5
6: 2345: respawn: /sbin/mingetty tty6
# Run xdm in runlevel 5
x: 5: respawn: /etc/X11/prefdm-nodaemon
```

【问题 1】(3 分)

在一台 256 MB RAM 的计算机上安装 Linux 系统,通常交换分区(swap)的大小合理的设置应该为(1)。

Linux 通过字母和数字的组合来标识硬盘分区, /dev/sdb5 指第 2 个硬盘的第(2)个逻辑分区,该硬盘是(3)类型。

【问题 2】(2 分)

Red Flag Desktop 4.0 安装光盘提供了四个安装引导软盘的映像文件,分别是 boot.img、boothd.img、bootnet.img 和 pcmcia.img。如果从硬盘安装 Linux,那么需要选择的引导文件是(4)。在 Linux 安装完成之后,(5)目录保存系统引导文件。

(5) 空缺处供选择的答案:

- A. / B. /root C. /home D. /boot

【问题 3】(2 分)

init 进程是系统中所有进程的父进程,init 进程的主要任务是按照 inittab 文件所提供的信息创建出完成通常操作所需的子进程,这些操作包括设置机器名、检查和安装磁盘及文件系统、启动系统日志、配置网络接口、启动网络和邮件服务、启动打印服务等。由此可见,inittab 文件存放在(6)目录中。根据试题说明中 inittab 文件的配置内容,系统引导成功后将工作在(7)状态。

(6) 空缺处供选择的答案:

- A. /etc B. /boot C. /sbin D. /root

(7) 空缺处供选择的答案:

- A. 单用户字符模式 B. 多用户字符模式
C. 单用户图形模式 D. 多用户图形模式

【问题 4】(2 分)

Linux 系统运行级别 3 工作在(8)状态。在系统控制台,(9)用<Ctrl>+<Alt>+<Delete>组合键来重新引导服务器。

(8) 空缺处供选择的答案:

- A. 单用户字符模式 B. 多用户字符模式
C. 单用户图形模式 D. 多用户图形模式

(9) 空缺处供选择的答案:

- A. 允许 B. 不允许
C. 禁止 D. 只能在 RPM 环境

【问题 5】(2 分)

假设 root 用户执行“init 0”命令,系统将会(10)。若 root 用户执行“ps aux | grep init”命令,得到 init 的 PID 是(11)。

(10) 空缺处供选择的答案:

A. 暂停 B. 关机 C. 重新启动 D. 初始化

(11) 空缺处供选择的答案:

A. 0 B. 1 C. 2 D. 3

【问题 6】(2 分)

根据上述 inittab 文件的内容, 系统在引导过程结束前, 至少还要执行(12)进程。

A. rc.sysinit B. rc.sysinit、rc 0、rc 1、rc 2、rc 3、rc 4、rc 5 和 rc 6
C. rc.sysinit 和 rc 5 D. rc 0、rc 1、rc 2、rc 3、rc 4、rc 5 和 rc 6

【问题 7】(2 分)

在任何时候, 可以在文件 inittab 中添加新的记录项。重新检查 inittab 文件时, 可以通过(13)命令使 init 进程立即重新读取并处理文件 inittab。

二、要点解析

【问题 1】(3 分)

这是一道要求读者掌握 Linux 系统基本安装常识的填空题。在实际应用中, 需要根据实际情况创建、修改或删除磁盘分区。Linux 系统通过字母和数字的组合来标识硬盘分区, 如表 5-1 所示。

表 5-1 Linux 系统硬盘分区各字段意义表

前两个字母	分区所在设备的类型	hd: IDE 硬盘 sd: SCSI 硬盘
第 3 个字母	分区在哪个设备上	hda: 第 1 块 IDE 硬盘 hdb: 第 2 块 IDE 硬盘 sdc: 第 3 块 SCSI 硬盘
数字	分区的次序	数字 1~4 表示主分区或扩展分区, 逻辑分区从 5 开始
例如: /dev/hda3 是指第 1 个 IDE 硬盘上的第 3 个主分区或扩展分区, 该硬盘是 IDE 类型		

由表 5-1 可知, /dev/sdb5 是指第 2 个 SCSI 硬盘上的第 1 个逻辑分区, 该硬盘是 SCSI 类型。另外需要注意的是, 如果硬盘上没有分区, 则一律不加数字, 代表整块硬盘。

对于题目中交换分区大小的合理设置问题, 需要先清楚硬盘分区的目的是在硬盘上为系统分配一个或几个确定的位置。Linux 系统支持多分区结构, 每一部分可以存放在不同的磁盘或分区上。各分区的功能见表 5-2。

表 5-2 Linux 系统各分区功能表

序号	分区	功 能
1	/	整个系统的基础(必备)
2	swap	操作系统的交换空间(必备)
3	/boot	在根下创建, 用来单独保存操作系统的内核和系统引导文件
4	/usr	用来保存系统软件
5	/home	包含所有用户的主目录, 可保存几乎所有的用户文件

续表

序号	分区	功 能
6	/var/log	系统日志记录分区
7	/tmp	用来存放临时文件, 对于多用户系统或者网络服务器是有必要的
8	/bin	存放标准系统实用程序
9	/dev	存放设备文件
10	/opt	存放可选的安装的软件
11	/sbin	存放标准系统管理文件

通常安装 Linux 系统至少需要一个根文件系统分区和一个交换分区。因为根分区(/)是 Linux 根文件系统驻留的地方。交换分区(swap)是用来支持虚拟内存的交换空间。当没有足够的内存来处理系统数据时, 就要使用交换分区的空间。交换分区的大小建议设置为计算机内存的 2 倍。

另外介绍两个常用的分区工具: 一个是 Disk Druid, 它是 Red Hat Linux 提供的硬盘管理工具。它可以根据用户的要求创建和删除硬盘分区, 另外还可以为每个分区管理载入点, 这是一个不错的分区软件, 建议读者使用; 另一个是 Fdisk, 它是传统的 Linux 硬盘分区工具, 比 Disk Druid 更强大, 使用更加灵活。但是 Fdisk 要求用户对硬盘分区有一定经验, 并能够适应且读懂简单的文本界面。

【问题 2】(2 分)

这也是一道要求读者掌握 Linux 系统基本安装常识的填空题。

Red Flag Linux Desktop 4.0 安装光盘的 /images 目录下提供了四个安装引导软盘的映像文件, 分别是 boot.img、boothd.img、bootnet.img 和 pcmcia.img。其中, boot.img 是使用光盘安装时负责引导安装的映像文件; boothd.img 是使用硬盘安装时负责引导安装的映像文件; bootnet.img 是使用网络安装(如 NFS 方式)时的引导映像文件; pcmcia.img 是为使用 PCMCIA 卡的用户准备, 用来支持笔记本电脑的安装引导。题目中要求从硬盘安装 Linux, 因此(4)空缺处应该选择 boothd.img 映像文件。

查阅表 5-2 Linux 系统各分区功能表可知, Linux 系统引导文件保存在 /boot 目录中, 因此(5)空缺处应选择 B 选项“/root”。

【问题 3】(2 分)

对于(6)空缺处所填写的内容, 由于 inittab 是系统 init 进程运行的配置文件, 因此应存放在 /etc 目录中。选项 B“/boot”目录主要用于存放内核以及启动所需的文件等; 选项 C“/sbin”目录用于存放系统管理程序; 选项 D“/root”目录为系统超级用户的主目录。

inittab 文件是以行为单位的描述性(非执行性)文本, inittab 文件中每一记录都从新的一行开始, 每个记录项最多可有 512 个字符, 每一个指令行都具有以下格式:

id:rstate:action:process

其中, “id”字段是一个最多由 4 个字符组成的字符串, 用来惟一地标识表项。

“rstate”字段用于定义该记录项被调用时的运行级别, rstate 可以由一个或多个运行级别构成, 也可以是空, 空则代表运行级别 0~6。当请求 init 改变运行级别时, 那些 rstate

字段中不包括新运行级别的进程将收到 SIGTERM 警告信号，并且最后被杀死。

“action”字段用于通知 init 执行什么操作，即如何处理 process 字段指定的进程，action 字段允许的值及对应的功能见表 5-3 所示。

表 5-3 action 字段允许的值及对应的功能表

action 字段值	功 能
respawn	如果 process 字段指定的进程不存在，则启动该进程，init 不等待处理结束，而是继续扫描 inittab 文件中的后续进程；当该进程已被终止，init 将会重新启动它；如果该进程已存在，则不进行任何操作
wait	启动 process 字段指定的进程，并等到处理结束才去处理 inittab 中的下一记录项
once	启动 process 字段指定的进程，不等待处理结束就去处理下一记录项；当该进程已被终止时，则不再重新启动它；在进入新的运行级别时，如果该进程仍在运行，init 也不重新启动它
boot	只有在系统启动时，init 才处理这样的记录项，启动相应进程，并不等待处理结束就去处理下一个记录项。当该进程已被终止时，系统也不重启它
bootwait	系统启动后，当第一次从单用户模式进入多用户模式处理这样的记录项时，init 启动该进程，并且等待它的处理结束，然后再进行下一个记录项的处理。当该进程已被终止时，系统也不重启它
powerfail	当 init 接到断电的信号(SIGPWR)时，处理指定的进程
powerwait	当 init 接到断电的信号(SIGPWR)时，处理指定的进程，并且等到处理结束才去检查其他的记录项
off	如果指定的进程正在运行，init 将先发 SIGTERM 警告信号，接着再向它发出信号 SIGKILL 强制其结束之前等待 5 秒。如果该进程不存在，则忽略这一项
ondemand	功能基本与“respawn”相同，不同之处在于与具体的运行级别无关，只用于 rstate 字段是 a、b、c 的记录项
sysinit	指定的进程在访问控制台之前执行，该记录项仅用于对某些设备的初始化，目的是为了使 init 在这样的设备上向用户提问有关运行级别的问题，init 需要等待进程运行结束后才继续
initdefault	指定一个默认的运行级别，只有当 init 一开始被调用时才扫描这一项；如果 rstate 字段指定了多个运行级别，则其中最大的数字是默认的运行级别；如果 rstate 字段是空的，init 认为字段是 0123456，于是进入级别 6，从而陷入了一个循环；如果 inittab 文件中没有包含 initdefault 的记录项，则在系统启动时请求用户为它指定一个初始运行级别

“process”字段用于指明某一进程，该进程可以是任意的守候进程、可执行脚本或程序。

在 inittab 配置文件清单中已给出以下提示信息：

```
# 0 - halt (Do NOT set initdefault to this)      //关机，请不要让默认 init 进程为 0
# 1 - Single user mode                            //单用户模式
# 2 - Multiuser, without NFS (The same as 3, if you do not have networking)
                                                //没有 NFS 的多用户模式
# 3 - Full multiuser mode                        //多用户模式
# 4 - unused                                       //保留
# 5 - X11                                          //X Window 模式
# 6 - reboot (Do NOT set initdefault to this)    //重新启动，请不要让默认 init 进程为 6
```

根据 inittab 文件清单中第 1 个配置行“id:5:initdefault:”可推理出，当 Linux 系统引导成功后，系统设置默认的运行级别是 5，即系统引导成功后将工作在多用户图形模式。

【问题 4】(2 分)

由 inittab 文件清单中注释语句“# 3 - Full multiuser mode”可知，Linux 系统运行级别 3 工作在多用户字符模式。同理可从该清单中得知，Linux 系统运行级别 1 工作在单用户模式；运行级别 5 工作在多用户图形模式。

inittab 文件清单中配置行“ca::ctrlaltdel:/sbin/shutdown -t3 -r now”指明允许 init 将〈Ctrl〉+〈Alt〉+〈Delete〉组合键(产生 SIGINT 中断信号)功能转移到调用关机脚本，立即进行重新引导(-r 选项)。

【问题 5】(2 分)

“init 0”是指以等级 0 运行 init 程序。由注释行“0 - halt (Do NOT set initdefault to this)”可知，root 用户执行“init 0”后，系统将会关机。

由于进行系统初始化的子进程都由 init 进程创建，因此 init 进程也称为系统初始化进程。可见 init 进程是系统所有进程的起点，是系统运行的第 1 个进程，它的进程号(PID)是 1。通过命令“ps aux | grep init”可查看到该 PID 号。

【问题 6】(2 分)

inittab 文件清单中配置行“si::sysinit:/etc/rc.d/rc.sysinit”指明 init 在调用任何规定运行级别的脚本之前，需要运行 rc.sysinit 脚本，然后运行与 7 个常用级别对应的一行命令脚本，即 rc 5(配置清单中“id:5:initdefault:”指出默认运行级别是 5)。

【问题 7】(2 分)

在任何时候，可以在文件 inittab 中添加新的记录项，级别 Q/q 不改变当前的运行级别。重新检查 inittab 文件时，可以通过命令 init Q 或 init q 使 init 进程立即重新读取并处理文件 inittab。

三、参考答案

表 5-4 给出了本案例问题 1 至问题 7 的参考答案，供读者练习时参考，以便查漏补缺。读者也可依照所给出的评分标准得出自己的测试分数，从而大致评估自己对这一方面知识点的掌握程度。

表 5-4 参考答案及评分标准表

问题与分值	参考答案及评分标准	自评分
【问题 1】 (3 分)	(1) 512 MB (2) 1 (3) SCSI(每空 1 分)	
【问题 2】 (2 分)	(4) boothd. img (5) B 或 /root(每空 1 分)	
【问题 3】 (2 分)	(6) A 或 “/etc” (7) D 或 “多用户图形模式” (每空 1 分)	
【问题 4】 (2 分)	(8) B 或 “多用户字符模式” (9) A 或 “允许” (每空 1 分)	
【问题 5】 (2 分)	(10) B 或 “关机” (11) B 或 “1” (每空 1 分)	
【问题 6】 (2 分)	(12) C 或 “rc. sysinit 和 rc 5”(2 分)	
【问题 7】 (2 分)	(13) init Q 或 init q(2 分)	

5.2.2 案例 2 RPM 操作技术

一、案例描述

认真阅读下列有关 Linux RPM 基本操作技术的说明，根据要求回答问题 1 至问题 7。
(15 分)

【说明】

软件包管理器(RPM)是一个免费开放打包系统，它可在 Red Hat Linux、其他 Linux 和 UNIX 系统上运行。对于终端用户来说，RPM 简化了系统的更新操作。RPM 软件包的安装、删除和升级等操作均可以使用简短的命令来完成；RPM 软件包还可以维护一个已安装软件包和它们的文件的数据库，因此，用户可以在 Linux 系统上使用强大的查询和校验功能。

【问题 1】(2 分)

不包括软件包建构，RPM 提供了哪些基本操作模式？

【问题 2】(3 分)

如需安装一个版本为 2.0、发行版本为 1、i386 体系的软件包，用户需先以根用户 root 身份登录，然后在 shell 提示符下键入(1)命令。

如果该软件包安装成功，Linux 系统则会输出如图 5-1 所示的提示信息。

Preparing	##### [100%]
1:foo	##### [100%]

图 5-1 软件包安装成功后的提示信息

如果键入(1)命令后,输出的提示信息如图 5-2 所示,说明用户在软件包安装过程中疏忽了什么问题?:

```
warning: V3 DSA signature: NOKEY, key ID 0352860f
```

图 5-2 软件包安装过程的 warning 提示信息

【问题 3】(2 分)

当以前用 RPM 安装的 foo-2.0-1.i386.rpm 软件包中某个文件被无意删除时,用户想通过安装该软件包中的初始配置文件来恢复,此时需在安装命令后带上(2)参数。

- A. --replacefiles
- B. --replacepkgs
- C. --redhatprovides
- D. --nodeps

【问题 4】(2 分)

使用 rpm -q 命令可查询用户所安装的 foo-2.0-1.i386.rpm 软件包的数据库。如果该命令后跟上(3)选项,还可用来显示被标记为配置文件的文件列表。

- A. -f
- B. -i
- C. -c
- D. -l

【问题 5】(2 分)

RPM 软件包可以使用隐私卫士 GnuPG 来签名,从而帮助用户鉴别所下载软件包的信任性。在成功安装 Red Hat Linux 操作系统后,GnuPG 软件已被默认安装。如果用户要使用 GnuPG 来校验从 Red Hat 内核小组收到的软件包,首先需要导入 Red Hat 的公钥。要导入这一公钥,需在 shell 提示符下执行以下命令:(4) /usr/share/rhn/RPM-GPG-KEY。

- A. rpm -K
- B. rpm-import
- C. rpm -V
- D. rpm-qi

【问题 6】(2 分)

假设 Linux 系统的 /usr/bin/paste 文件出了问题,用户想校验拥有该文件的软件包,但又不知道哪个软件包拥有此 paste 文件,此时,可使用(5)命令来诊断问题。

- A. rpm-Va
- B. rpm-ql /usr/bin/paste
- C. rpm-qa
- D. rpm-Vf /usr/bin/paste

【问题 7】(2 分)

为了确保内核的完整性及对它所支持的硬件的兼容性,Red Hat Linux 内核均由 Red Hat 内核小组定制建构。由于 Red Hat Linux 内核使用 RPM 格式打包,因而它们易于升级和校验。如果用户需要安装内核软件包,可以使用(6)命令来完成这一操作任务。

- A. LiLo
- B. install
- C. rpm -ivh
- D. rpm -Uvh

二、要点解析

【问题 1】(2 分)

对于本问题读者在解答时可以从题目中关键信息“RPM 软件包的安装、删除、升级等操作均可以使用简短的命令来完成;RPM 软件包还可以维护一个已安装软件包和它们的文件的数据库,因此,用户可以在 Linux 系统上使用功能强大的查询和校验功能”中提取而形成答案,即把这句提示信息中的关键词“安装”、“删除”、“升级”、“查询”和“校验”提取出来。以下给出这 5 种操作模式的基本操作命令:

① 安装软件包需先登录成为根用户，然后在 shell 提示符下键入以下命令：`rpm -ivh` <软件包名称>。

② 删除软件包需先登录成为根用户，然后在 shell 提示符下键入下面的命令：`rpm -e` <软件包名称>。

③ 因为 RPM 对软件包和配置文件执行智能升级，所以如果用户自己决定升级软件包，可在 shell 提示符下键入与安装操作相类似的命令：`rpm -Uvh` <软件包名称>；如果要刷新某个软件包，可在 shell 提示符下键入以下命令：`rpm -Fvh` <软件包名称>。

④ 使用 `rpm -q` 命令来查询所安装的软件包的数据库。

⑤ 使用 `rpm -V` 命令可校验某一软件包。

【问题 2】(3 分)

通常一个 RPM 软件包的名称由软件包名称、版本、发行版本和体系等四部分组成。例如：对于一个名为 `foo-2.0-1.i386.rpm` 的文件，该文件的软件包名称为 `foo`，版本为 `2.0`、发行版本为 `1`，体系为 `i386`。由图 5-1 所示的提示信息，第 1 行“Preparing”可知，该行表示操作系统预备安装该软件包的进度，即以“#”号来显示预备的进度；第 2 行中“1”表示现在正在安装第 1 个软件包，“foo”是所安装软件包的名称。因此先以根用户 `root` 身份登录，然后在 shell 提示符下键入命令 `rpm -ivh foo-2.0-1.i386.rpm` 即可安装 `foo-2.0-1.i386.rpm` 这个软件包。

从 RPMv 4.1 版本开始，在安装或升级软件包时会检查软件包的签名。如果软件包安装过程中签名校验失败，用户就会看到如图 5-3 第 1 行所示的包含“BAD”的错误信息。如果这一签名是只针对文件头的，用户会看到如图 5-3 第 2 行所示的包含“Header”、“BAD”的错误信息。如果用户没有安装合适的钥匙来校验签名，提示信息中就会包含“NOKEY”信息，如图 5-3 第 3 行所示。

1	error: V3 DSA signature: BAD, key ID 0352860f
2	error: Header V3 DSA signature: BAD, key ID 0352860f
3	warning: V3 DSA signature: NOKEY, key ID 0352860f

图 5-3 使用 `rpm-ivh` 命令可能出现的几种提示信息

【问题 3】(2 分)

如果用户试图安装的 `foo-2.0-1.i386.rpm` 软件包中包含已被另一个软件包或同一软件包的早期版本安装了的文件，则当用户执行 `rpm-ivh foo-2.0-1.i386.rpm` 命令后，就会看到图 5-4 第 1 行、第 2 行的提示信息。如果要使 RPM 管理器忽略这个错误，就可使用 `--replacefiles` 选项，即将安装命令更改为 `rpm -ivh --replacefiles foo-2.0-1.i386.rpm`。

当以前用 RPM 安装 `foo-2.0-1.i386.rpm` 软件包中的文件被无意删除了，用户想安装该软件包中的初始配置文件，此时可以使用 `--replacepkgs` 选项，键入命令 `rpm-ivh --replacepkgs foo-2.0-1.i386.rpm`，告诉 RPM 在该软件包已安装的情况下仍打算覆盖安装同一版本的软件包。如没有使用这一选项，将会看到图 5-4 第 3 行、第 4 行的提示信息。

由于 Linux 系统是一个开源系统，因此某些软件包之间可能存在着“依赖”的安装关系，即安装某个软件包之前需要先安装其他软件包。当某两个软件包都安装成功时，就会看到类似图 5-4 第 5~7 行的提示信息。如果用户已经安装了 `rpmdb-redhat` 软件包，就可以使用 `--redhatprovides` 选项来判定哪个软件包中包含了所需的依赖文件。

如果在安装命令后带上 `-nodeps` 选项,就可强制安装这些存在“依赖”关系的软件包,但所安装的软件包可能不能够正确运行。

1	Preparing...	##### [100%]
2	file /usr/bin/foo from install of foo-2.0-1 conflicts with file from package bar-2.0.20	
3	Preparing...	##### [100%]
4	package foo-2.0-1 is already installed	
5	Preparing...	##### [100%]
6	1:foo	##### [50%]
7	2:bar	##### [100%]

图 5-4 使用 `rpm-ivh` 命令可能出现的几种提示信息

【问题 4】(2 分)

使用 `rpm -q` 命令可查询所安装的 `foo-2.0-1.i386.rpm` 软件包的数据库,例如键入 `rpm -q foo` 命令会显示出 `foo-2.0-1`,即所安装的软件包 `foo` 的名称、版本和发行号码。

指定被查询的软件包要显示哪些信息可以有多种方法。其中, `-c` 选项可用来显示被标记为配置文件的文件列表。如果要求以熟悉的 `ls -l` 格式来显示文件列表,可以在“`-c`”、“`-d`”选项之后添加“`-v`”来完成。表 5-5 整理出一些 `rpm -q` 查询命令后常带的选项。

表 5-5 查询命令后常用的选项

选项参数	功 能
<code>-a</code>	查询所有已安装的软件包
<code>-c</code>	显示被标记为配置文件的文件列表
<code>-d</code>	显示被标记为文档(man 页、info 页等)的文件列表
<code>-f <file></code>	查询拥有<file>的软件包,注意必须指定文件的完整路径,如 <code>/usr/bin/ls</code>
<code>-i</code>	显示软件包信息,如名称、描述、发行版本、大小、制造日期和生产商等
<code>-l</code>	显示软件包所含的文件列表
<code>-p <packagefile></code>	查询<packagefile>软件包
<code>-s</code>	显示软件包中所有文件的状态

【问题 5】(2 分)

RPM 软件包可以使用隐私卫士 GnuPG 来签名,从而帮助用户鉴别所下载软件包的信任性。在成功安装 Red Hat Linux 操作系统后, GnuPG 软件已被默认安装。如果用户要使用 GnuPG 来校验从 Red Hat 内核小组收到的软件包,首先需要导入 Red Hat 的公钥。要导入这一公钥,需在 shell 提示符下执行 `rpm --import /usr/share/rhn/RPM-GPG-KEY` 命令。执行 `rpm -qa gpg-pubkey *` 命令可显示用来校验 RPM 而安装的钥匙列表。对于 Red Hat 公钥,在以上两条操作命令的输出信息中应包括“`gpg-pubkey-db42a60e-37ea5438`”信息。接着使用 `rpm -qi gpg-pubkey-db42a60e-37ea5438` 命令可显示 Red Hat 公钥的一些细节信息。

导入了 Red Hat 的 GnuPG 公钥后,要检查 RPM 文件的 GnuPG 签名,可使用 rpm -K <rpm-file>命令,其中,<rpm-file>是指具体 RPM 软件包的名称。如果签名校验成功,即该软件包没有被损坏, Linux 系统就会输出“md5 gpg OK”的提示信息。

与 rpm-K 校验命令相类似的校验命令是 rpm-V。但 rpm-V 命令主要用于校验每个文件的大小、MD5 值、权限、类型、所有者及组群关系。例如,用户怀疑 foo 软件包的数据库已被损坏,可使用 rpm-V foo 命令,RPM 将校验所有在 foo 软件包内的文件是否和初始安装时的同一文件一致。另外还有几条常用的校验命令,rpm-Vf /bin/vi 命令可校验包含某一特定文件的软件包;rpm-Vp foo-2.0-1.i386.rpm 命令可根据 RPM 软件包文件来校验安装后的 foo-2.0-1 软件包;如要校验所有安装了的软件包,可使用 rpm-Va 命令。

最后说明一下使用以上校验命令的输出信息。如果软件包被校验成功, Linux 系统将不会有提示信息输出。如果软件包校验失败, Linux 系统将输出 8 个代表文件属性的字符串,其中每一个字符都代表一种文件属性和 RPM 数据库中记录的属性的比较结果,表 5-6 所示是校验结果提示信息中常见的一些字符及其意义。

表 5-6 校验结果常见的提示字符

字符	意 义	字符	意 义
• (点)	该文件属性测试通过	M	模式(包括权限和文件类型)测试失败
5	MD5 校验和测试失败	S	文件大小测试失败
c	该文件是一个配置文件	T	文件修改时间
D	设备测试失败	U	所有者测试失败
G	组群测试失败	?	该文件是一个不可读文件
L	符号链接测试失败		

【问题 6】(2 分)

RPM 管理器为管理系统文件、诊断和修正问题提供了便捷的通道。例如,当用户不小心删除了一些 Linux 文件,却不能肯定删除了哪些具体文件,此时可使用 rpm-Va 命令来校验整个系统以发现缺少哪些文件。当 Linux 系统的/usr/bin/paste 文件出了问题,用户想校验拥有该文件的软件包,但又不知道哪个软件包拥有此 paste 文件,此时可使用 rpm -Vf /usr/bin/paste 命令来诊断问题。

【问题 7】(2 分)

为了确保内核的完整性及对其所支持硬件的兼容性, Red Hat Linux 内核均由 Red Hat 内核小组定制构建。在内核被发行之之前,它经过了一系列严格的质量保证测试。

由于 Red Hat Linux 内核使用 RPM 格式打包,因而它们易于升级和校验。如果用户需要安装内核软件包,则可以使用 rpm -ivh 命令来完成这一操作任务。

三、参考答案

表 5-7 给出了本案例问题 1 至问题 7 的参考答案,供读者练习时参考,以便查漏补缺。读者也可依照所给出的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

表 5-7 参考答案及评分标准

问题与分值	参考答案及评分标准	自评分
【问题 1】(2 分)	不包括软件包建构, RPM 有五种基本操作模式: 安装、删除、升级、查询和校验 (答对两种模式得 1 分, 全对得 2 分)	
【问题 2】(3 分)	(1) rpm -ivh foo-2.0-1.i386.rpm(2 分) 没有安装合适的钥匙来校验签名(1 分)	
【问题 3】(2 分)	(2) B 或 --replacekgs	
【问题 4】(2 分)	(3) C 或 --c	
【问题 5】(2 分)	(4) B 或 rpm --import	
【问题 6】(2 分)	(5) D 或 rpm -Vf /usr/bin/paste	
【问题 7】(2 分)	(6) C 或 RPM-ivh	

5.2.3 案例 3 内核移植与编译

一、案例描述

请认真阅读下列有关于 Linux 内核移植与编译技术的说明, 根据要求回答问题 1 至问题 6。(15 分)

【说明】

所谓 Linux 移植, 就是把 Linux 操作系统针对具体的目标平台做必要的改写之后, 安装到该目标平台使其正确地运行起来。Linux 内核移植大致可以归纳成以下几个步骤:

- ① 准备工作, 下载 Linux 内核源代码和编译器源代码等;
- ② 建立交叉编译环境;
- ③ 制作 Boot Loader;
- ④ 修改和编译内核;
- ⑤ 制作文件系统;
- ⑥ 编写相应的设备驱动;
- ⑦ 编写应用程序。

在 Linux 启动的第一阶段, 内核与体系结构相关部分(arch 目录下)首先执行, 它会完成硬件寄存器设置、内存映射等初始化工作, 然后再把控制权转给内核中与系统结构无关的部分。在 arch 目录中可以看到许多子目录, 它们往往是用芯片命名的, 表示针对该芯片体系结构的代码。为 ARM 系列芯片编译内核, 就应修改 arm 目录下的相关文件。在 /arch/arm/boot/bootp 目录下可以找到一个 init.S 的文件。

init.S 文件的源代码如下:

```
/* linux/arch/arm/boot/bootp/init.S
 * Header file for splitting kernel + initrd. Note that we pass
 * r0 through to r3 straight through. */
```

```

        .section .start, #alloc, #execinstr
        .type _entry, #function
_entry:      adr r10, initdata
            ldr r11, initdata
            sub r11, r10, r11          @ work out exec offset
            b splitify
        .size _entry, . - _entry

        .type initdata, #object
initdata:    .word initdata          @ compiled address of this
        .size initdata, . - initdata

        .text
splitify:    adr r13, data
            ldmbia r13!, {r4 - r6}    @ move the initrd
            add r4, r4, r11          @ correction
            bl move

            ldmbia r13!, {r4 - r6} @ then the kernel
            mov r12, r5
            add r4, r4, r11 @ correction
            bl move

/* Setup the initrd parameters to pass to the kernel. This can either be
 * passed in via a param_struct or a tag list. We spot the param_struct
 * method by looking at the first word; this should either indicate a page
 * size of 4K, 16K or 32K. */
            ldmbia r13, {r4 - r8} @ get size and addr of initrd
                                @ r5 = ATAG_INITRD
                                @ r6 = initrd start
                                @ r7 = initrd end
                                @ r8 = param_struct address
            ldr r9, [r8, #0] @ no param struct?
            teq r9, #0x1000 @ 4K?
            teqne r9, #0x4000 @ 16K?
            teqne r9, #0x8000 @ 32K?
            beq param_struct

            ldr r9, [r8, #4] @ get first tag
            teq r9, r4
            bne taglist      @ ok, we have a tag list

/* We didn't find a valid tag list - create one.
            str r4, [r8, #4]

```

```

    mov r4, #8
    str r4, [r8, #0]
    mov r4, #0
    str r4, [r8, #8]

/* find the end of the tag list, and then add an INITRD tag on the end.
 * If there is already an INITRD tag, then we ignore it; the last INITRD
 * tag takes precedence. */
taglist: ldr r9, [r8, #0]      @ tag length
        teq r9, #0            @ last tag?
        addne r8, r8, r9
        bne taglist

        mov r4, #16           @ length of initrd tag
        mov r9, #0            @ end of tag list terminator
        stmia r8, {r4, r5, r6, r7, r9}
        mov pc, r12           @ call kernel

/* We found a param struct. Modify the param struct for the initrd
param_struct: add r8, r8, #16 * 4
        stmia r8, {r6, r7}    @ save in param_struct
        mov pc, r12           @ call kernel

move:   ldmia r4!, {r7 - r10}  @ move 32 - bytes at a time
        stmia r5!, {r7 - r10}
        ldmia r4!, {r7 - r10}
        stmia r5!, {r7 - r10}
        subs r6, r6, #8 * 4
        bcs move
        mov pc, lr

data:   .word initrd_start
        .word initrd_addr
        .word initrd_len

        .word kernel_start
        .word kernel_addr
        .word kernel_len

        .word 0x54410001      @ r4 = ATAG_CORE
        .word 0x54420005      @ r5 = ATAG_INITRD
        .word initrd_addr     @ r6
        .word initrd_len      @ r7

```



```
. word params          @ r8

.type kernel_start, # object
.type initrd_start, # object
```

- 【问题 1】(3 分)
最基本的嵌入式 Linux 系统由哪三个基本元素组成?
- 【问题 2】(3 分)

交叉编译环境的建立最重要的就是要有一个交叉编译器。交叉编译器的本质是把某种以数字和符号为内容的高级编程语言转换成机器语言指令的集合。编译器的工作阶段示意图如图 5-5 所示, 请将图中(1)~(3)空缺处的内容填写完整。

- 【问题 3】(2 分)
目前交叉编译技术主要有哪两种典型的实现模式?
- 【问题 4】(4 分)

init. S 是引导 uCLinux 内核在 Arm 平台上启动的初始化代码。它定义了一个全局符号_start, 它既是默认的起始地址, 同时也是整体内核二进制镜像的起始标志。init. S 文件的后缀名“. S”表示什么意义? 仔细阅读试题给出的 init. S 文件源代码, 请在 150 字以内归纳总结出该文件主要完成的功能。

- 【问题 5】(1 分)

Linux 内核的编译菜单通过(4)经不同脚本解释器产生 .config。它包括“make config”、“make menuconfig”和“make xconfig”等版本。

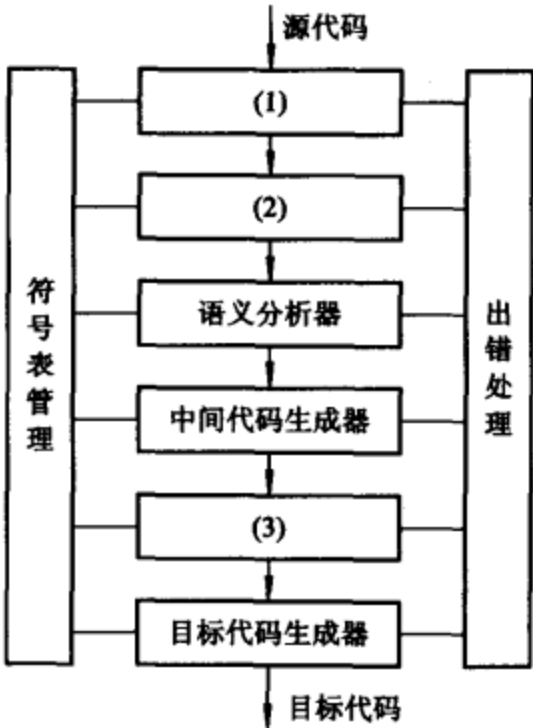


图 5-5 编译器的工作阶段示意图

- 【问题 6】(2 分)

在完成内核的裁减之后, 内核的编译就是一个非常简单的过程。在正式编译内核之前执行(5)命令可彻底清除相关依赖, 保证没有不正确的.o 文件存在。(6)命令是最终的编译命令。

(5) 空缺处供选择的答案:

- A. make clean B. make dep
C. make del D. make mrproper

(6) 空缺处供选择的答案:

- A. make realclean B. make zImage
C. make install D. make save config

二、要点解析

【问题 1】(3 分)

这是一道要求读者掌握最小 Linux 系统组成的基本常识题。最基本的嵌入式 Linux 系统由系统引导程序、Linux 系统内核和初始化过程等 3 个基本元素组成。其中:

- ① 系统引导程序用于完成机器加电后的系统定位引导;
- ② Linux 系统内核为嵌入式应用提供一个软件环境, 为应用程序完成基本的底层的资源管理工作;
- ③ 初始化过程用于完成基本的初始化。

由此可知, 为使这个最小嵌入式系统具有一定的实用性, 还需加上硬件的驱动程序及一个或几个应用进程以提供必要的功能支持。如果应用于比较复杂的工作环境, 则可能还需要添加一个可以在 ROM 或 RAM 中使用的文件系统、TCP/IP 网络协议栈等。如果在 PDA 领域中应用, 则需要加上一个 GUI 支持。

【问题 2】(3 分)

这是一道要求读者掌握编译程序工作过程的基本常识题。本题所涉及的知识点如下:

① 所谓交叉编译, 是指利用运行在某机器上的编译器编译某个源程序生成在另一台机器上运行的目标代码的过程。编译器的生成依赖于相应的函数库, 而这些函数库又得依靠编译器来编译。通常将编译程序的工作过程分为六个阶段, 如图 5-6 所示。每个阶段的操作在逻辑上是紧密相连的, 将源程序从一种表示形式一步步转换成目标代码。

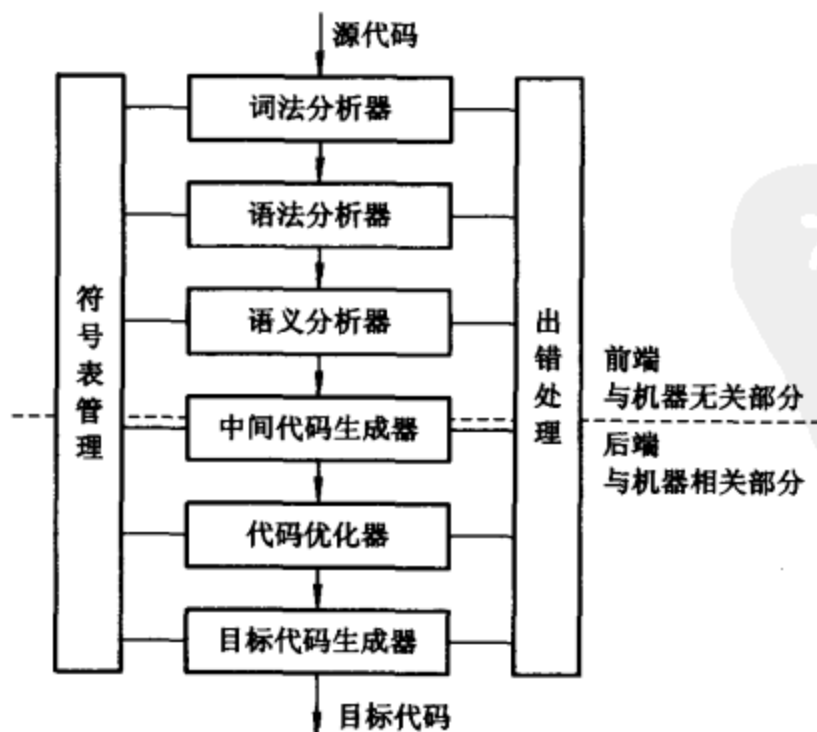


图 5-6 完整的编译器工作阶段示意图

② 表 5-8 归纳了图 5-6 中每个工作阶段实现的主要功能。

表 5-8 编译器六个工作阶段功能表

名 称	主 要 任 务	备 注
词法分析器	对源程序从前到后、从左到右逐个字符进行扫描，从中识别出一个个“单词”符号	这两个阶段的本质都是对源程序的结构进行分析
语法分析器	根据语言的语法规则将单词符号序列分解成各类语法单位，构造出相应的语法树	
语义分析器	对源程序进行类型分析并检查是否存在语义错误，收集类型信息供代码生成阶段使用	这两个阶段的依据是语言的语义规则
中间代码生成器	根据语义分析的输出生成与具体机器无关的中间代码	
代码优化器	优化中间代码在时间和空间上的浪费问题	依据是程序的等价变换规则
目标代码生成器	把中间代码变换成特定机器上的绝对指令码、可重定位的指令代码或汇编指令代码	此阶段工作与具体的机器相关
符号表管理	记录源程序中各个符号的必要信息，以辅助语义的正确性检查和代码生成	
出错处理	处理源程序中的静态错误和动态错误	

③ 由以上分析可知，(1)空缺处的内容应填入“词法分析器”，(2)空缺处的内容需填入“语法分析器”，(3)空缺处可填入“代码优化器”等内容。

【问题 3】(2 分)

这是一道要求读者掌握交叉编译技术实现模式的常识题。

目前交叉编译技术主要有 Java 模式和 GNU GCC 模式这两种典型的实现模式。其中，Java 模式即 Java 的字节码编译技术。其最大特点是引入一个自定义的虚拟机(JVM)，所有 Java 源程序都会被编译成在这个虚拟机上才能执行的“目标代码”——字节码。GNU GCC 模式通过 Cross GCC 工具集直接生成目标平台的目标代码，从而能够直接在目标平台上运行。它使用 RTL 寄存器传递语言对目标平台的指令进行描述，从而保证编译过程与具体硬件平台的无关性。相对来说，该模式比 Java 模式更为优化，效率更高。

【问题 4】(4 分)

这是一道要求读者掌握 Linux 系统内核源码——init. S 文件的程序阅读理解题。本试题的解答思路如下：

(1) init. S 文件的后缀“. S”表示它是一个汇编语言文件。本试题给出的 init. S 是用 ARM 汇编写成的。

(2) 由 init. S 文件中给出的注释内容可知，init. S 文件主要完成以下功能：

① 定义数据段、代码段、bbs(未初始化数据段)起始地址变量，并对 bbs 段进行初始化；

- ② 设置寄存器以初始化系统硬件；
- ③ 关闭中断；
- ④ 初始化 LCD 显示；
- ⑤ 将数据段数据复制到内存；
- ⑥ 跳转到内核起始函数 `start_kernel`，继续执行；
- ⑦ 修改相关的主寄存器。

其中，初始化设置的寄存器应根据具体的硬件平台，参考相应的芯片手册。通常要做修改的寄存器有：片选组基地址寄存器、DRAM 存储配置寄存器、DRAM 片选寄存器、中断屏蔽寄存器等。

【问题 5】(1 分)

这是一道要求读者掌握 Linux 内核裁减的基本常识题。本试题所涉及的知识点如下：

- ① Linux 内核的裁减是一个相对简单的过程，主要工作是对配置菜单的简单选择，但内核配置菜单本身结构庞大，内容复杂，做出选择前需要了解各个配置选项的作用。
- ② Linux 内核的编译菜单通过 `Config.in` 由不同脚本解释器产生 `.config`。
- ③ `make config` 版本的 Linux 内核的编译菜单主要以命令行方式进行功能配置。
- ④ `make menuconfig` 版本的 Linux 内核的编译菜单主要以菜单方式进行配置，且配置稳定可靠。
- ⑤ `make xconfig` 版本的 Linux 内核的编译菜单以基于图形库菜单方式进行配置，菜单结构清晰，配置界面友好。

【问题 6】(2 分)

这是一道要求读者掌握 Linux 系统内核编译命令的基本常识题。本题的解答思路如下：

① 对于(5)空缺处供选择的答案中，选项 A“`make clean`”是一条在正式编译内核之前先把环境给清理干净了的命令，但它不能实现彻底清除相关依赖，保证没有不正确的 `.o` 文件存在。

选项 B“`make dep`”是一条编译相关依赖文件的命令。

Linux 系统内核的编译命令中没有选项 C“`make del`”这一条命令。

选项 D“`make mrproper`”命令用于彻底清除相关依赖，保证没有不正确的 `.o` 文件存在。另外，命令“`make realclean`”也用于彻底清除相关依赖，保证没有不正确的 `.o` 文件存在。

因此(5)空缺处的答案应选择选项 D(`make mrproper`)。

② 对于(6)空缺处供选择的答案中，选项 A“`make realclean`”用于彻底清除相关依赖，保证没有不正确的 `.o` 文件存在。

选项 B“`make zImage`”命令就是最终的编译命令，有时可以直接用 `make(2.6.X 版本上)` 或 `make bzImage`(计算机编译大内核时使用)。

选项 C“`make install`”命令可以把相关文件拷贝到默认的目录。由于具体的内核安装还需要手工进行，因此在给嵌入式设备编译时可以省略该步骤。

Linux 系统内核的编译命令中没有选项 D“`make save config`”这一条命令。

由以上分析可知，(6)空缺处的答案需选择选项 B(`make zImage`)。

三、参考答案

表 5-9 给出了本案例问题 1 至问题 6 的参考答案,供读者练习时参考,以便查漏补缺。读者也可依照所给出的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

表 5-9 参考答案及评分标准

问题与分值	参考答案及评分标准	自评分
【问题 1】 (3 分)	① 系统引导程序 ② Linux 系统内核 ③ 初始化过程(每小点 1 分)	
【问题 2】 (3 分)	(1) 词法分析器 (2) 语法分析器 (3) 代码优化器(每空 1 分)	
【问题 3】 (2 分)	① Java 模式 ② GNU GCC 模式(每小点 1 分)	
【问题 4】 (4 分)	“S”表示该文件是一个汇编语言文件(0.5 分) init.S 文件主要完成以下功能: ① 定义数据段、代码段、未初始化数据段(bbs)起始地址变量,并对 bbs 段进行初始化 ② 设置寄存器以初始化系统硬件 ③ 关闭中断 ④ 初始化 LCD 显示 ⑤ 将数据段数据复制到内存 ⑥ 跳转到内核起始函数 start_kernel,继续执行 ⑦ 修改相关的主寄存器(每小点 0.5 分)	
【问题 5】 (1 分)	Config.in(1 分)	
【问题 6】 (2 分)	(5) D(make mrproper) (6) B(make zImage)(每空 1 分)	

5.2.4 案例 4 Samba 配置技术

一、案例描述

认真阅读下列有关 Linux 操作系统中 Samba 配置技术的说明,根据要求回答问题 1 至问题 7。(15 分)

【说明】

嵌入式文件系统的设计目标包括使用简单方便、实时响应、安全可靠、接口标注的开放性、可移植性、可伸缩性和可配置性、开放的体系结构、资源有效性、功能完整性和支持多种文件类型等功能。

嵌入式 Linux 文件系统有两条独立控制设备驱动的途径：一条通过设备驱动的接口；另一条通过文件管理器接口。

SMB(Server Message Block, 服务消息块)协议主要用于实现 Windows 和 Linux 操作系统中计算机之间共享打印机、串行接口和命名管道、邮件插槽等服务。Linux 操作系统的 Samba 是用来实现 SMB 功能的一种软件,它允许用户文件和打印机被网络中的所有操作系统共享。

Samba 的核心是两个守护进程 `smbd` 和 `nmdb` 程序,在服务器启动到停止期间持续运行。`smbd` 进程的作用是处理到来的 SMB 软件包,为使用该软件包的资源与 Linux 进行协商;`nmdb` 进程使其他主机(或工作站)能浏览 Linux 服务器。

图 5-7 所示是某实验室计算机利用 Samba 服务器来实现 Windows 98/2000/XP 及 Linux 等操作系统相互共享资源的网络结构图。

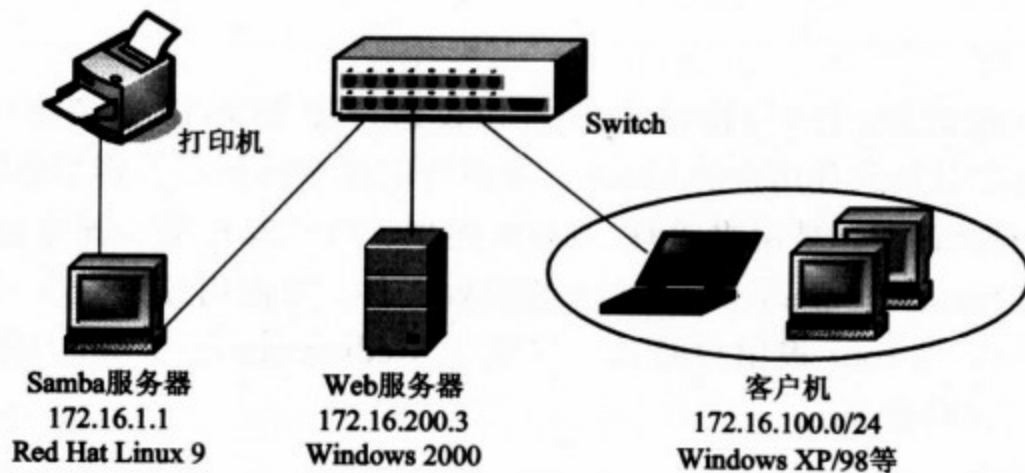


图 5-7 某实验室网络结构图

Samba 使用 `/etc/samba/smb.conf` 作为它的配置文件,以下是一个 `smb.conf` 文件的部分配置内容。

```

1 [global]
2  workgroup=WG_share
3  server string=%h
4  hosts allow=(4)
5  max disk size=(5)
6  max open file=8
7  log file=/var/log/samba/%m.log
8  max log size=(6)
9  security=(7)
10 password level=0
11 username level=0
12 encrypt passwords=yes
13 smb passwd file=/etc/samba/smbpasswd
14 username map=/etc/samba/smbusers
15 socket options=TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
16 dns proxy=no
17 [WG_Share]
```



```
18 comment=MY Share
19 path=/home/WG_Share
20 writable=yes
21 valid users=%U
22 create mode=0755
23 directory mode=0755
24 [printers]
25 path=/var/spool/samba
26 writeable=no
27 guest ok=yes
28 printable=yes
29 printer driver=HP color LaserJet 8550
.....
```

【问题 1】(3 分)

Samba 的工作原理是：让(1)和(2)这两种协议运行于 TCP/IP 通信协议之上，且通过 Windows 的 NETBEUI 协议让用户的 Linux 计算机可以在 Windows 的网络邻居上被看到。

Samba 服务器配置工具是用来管理 Samba 共享、用户以及基本服务器设置的图形化界面。它修改/etc/samba/smb.conf 目录中的配置文件。要使用该配置工具，用户必须运行 X-Window 系统，以 root 根用户身份且安装了 Redhat-config-samba RPM 软件包。在 shell 提示符下键入(3)命令。

【问题 2】(2 分)

在 smb.conf 配置文件中，hosts allow 语句用于设置允许访问共享网络的计算机 IP 地址。请结合图 5-7 给出的网络结构图，将 smb.conf 配置文件中(4)填写完整。

【问题 3】(2 分)

为了能尽可能多地详细记录系统日志和提高 Samba 服务器的利用率，要求对 smb.conf 日志文件的大小不做限制，并对 Samba 服务器的共享目录磁盘空间也不做大小限制。请将 smb.conf 文件中(5)和(6)空缺处的配置填写完整，以完成这一管理要求。

【问题 4】(2 分)

在题干所给出的 smb.conf 配置文件中，Samba 用户密码存放在哪个文件中？

在 smb.conf 配置文件中，security 语句用于定义 Samba 服务器的安全级别，有 share、user、server 和 domain 等四种常用级别。如果图 5-7 所示的网络允许任何用户都可以匿名方式访问 Samba 服务器上的共享资源，那么 smb.conf 文件中(7)空缺处应填入哪个配置参数？

【问题 5】(2 分)

本配置文件中，第 19~21 行语句完成什么配置功能？

【问题 6】(2 分)

对 smb.conf 文件的配置信息修改后，可执行(8)命令来测试该文件语法设置是否正确。测试通过后还需执行(9)命令来重启 smb 服务。

【问题 7】(2 分)

准备将 Samba 应用到开发板时，首先在准备烧写到 flash 的文件系统中创建以下目录：

```
/usr/local/samba  
/usr/local/samba/bin  
/usr/local/samba/var  
/usr/local/samba/private  
/usr/local/samba/lib
```

将编译后的 samba 源码目录下的/bin 子目录里的内容拷贝到(10)。接着配置 smb.conf 文件,添加、删除及修改 Samba 提供的多种服务,然后用 smbpasswd -a username 命令添加(11),接着运行 Samba 守护进程。

二、要点解析

【问题 1】(3 分)

Samba 使用 SMB(Server Message Block)协议通过网络连接来共享文件和打印机。它的工作原理是让 SMB 和 NETBIOS(Windows 操作系统网络邻居的通信协议)这两个协议运行于 TCP/IP 通信协议之上,并且使用 Windows 的 NETBEUI 协议让 Linux 计算机可以在网络邻居上被 Windows 计算机看到。Samba 具有如下功能:① 共享 Linux 磁盘给 Windows 操作系统;② 共享 Windows 操作系统磁盘给 Linux 机器;③ 共享 Linux 打印机给 Windows 操作系统;④ 共享 Windows 打印机给 Linux 机器。

Samba 服务器配置工具是用来管理 Samba 共享、用户以及基本服务器设置的图形化界面。它修改/etc/samba/smb.conf 配置文件。要使用该配置工具,用户必须运行 X-Window 系统,以根用户 root 身份登录且安装 redhat-config-sambaRPM 软件包,接着在 shell 提示符下键入 redhat-config-samba 命令。需要注意的是,Samba 服务器配置工具不显示允许用户在 Samba 服务器上查看他们的主目录的共享打印机或默认文件段。

【问题 2】(2 分)

在 smb.conf 配置文件中,hosts allow 语句用于设置允许访问共享网络的网段地址和主机 IP 地址。结合题干中“销售部计算机资源相互共享”及图 5-7 所示的网络结构可知,该共享网络允许客户机网段(172.16.100.0/24)和 Web 服务器(172.16.200.3/32)访问 Samba 服务器(172.16.1.1)的打印机资源。由此推出,host allow 语句空缺处应填入这些计算机的 IP 地址,即

hosts allow=172.16.100. 172.16.200.3 127.0.0.1

或

hosts allow=172.16.100. 172.16.200.3 172.16.1.1

进行该语句配置时需要注意三点:① 网段地址用“.”号来表达子网掩码中“0”的部分;② 各个 IP 地址或网段地址间用空格隔开;③ 记得将 Samba 服务器本机的 IP 地址添加进来。

【问题 3】(4 分)

Linux 系统 smb.conf 文件中,max log size 语句用于设置日志文件的最大容量,大小以 KB 为单位。当 size=0 时,表示对日志文件的大小没有限制。

max disk size 语句用于设置共享目录磁盘空间的大小,以 MB 为计算单位。当 size=0 时,表示对共享目录磁盘空间的大小没有限制。

【问题 4】(2 分)

在试题所给出的 smb.conf 配置文件中,第 12、13 行分别实现对密码进行加密和为 Samba 创建一个单独的口令文件的配置功能。

```
12 encrypt passwords=yes;    //对密码进行加密
```

```
13 smb passwd file=/etc/samba/smbpasswd;
```

```
//为 Samba 创建一个单独的口令文件/etc/samba/smbpasswd
```

这些功能的实现需在 Samba 配置文件被重启前先改变 Samba 口令文件的权限许可。要在 Red Hat Linux 用户系统上配置 Samba 使用加密口令,需经过以下的操作步骤:

① 为 Samba 创建一个单独的口令文件。注意要根据用户的现存/etc/passwd 文件来创建,可在 shell 提示符下键入以下命令:

```
cat /etc/passwd | mksmbpasswd.sh > /etc/samba/smbpasswd
```

② 改变 Samba 口令文件的权限许可。先以根用户 root 身份登录,然后在 shell 提示符下键入命令:

```
chmod 600 /etc/samba/smbpasswd
```

③ 在 smb.conf 配置文件中添加以下启用加密口令的语句。

```
encrypt passwords=yes;    对密码进行加密
```

```
smb passwd file=/etc/samba/smbpasswd;    为 Samba 创建一个单独的口令文件
```

④ 在 shell 提示符下键入 service smb restart 命令重启 smb 服务。

在 smb.conf 配置文件中,security 语句用于定义 Samba 的安全级别。Samba 的安全级别从低到高分别为 share、user、server 和 domain 四级,其中,后面三种安全级别都要求在用户的 Linux 机器上要有与服务器相同的系统账户,否则不能访问共享资源。

Samba 的各安全级别验证方式的比较见表 5-10。

表 5-10 Samba 的各安全级别验证方式比较

安全级别	配置参数	验证方式
最低	share	无安全性的级别,任何用户都可以不要用户名和口令就可访问共享服务器上的资源
低	user	要求用户在访问共享资源之前必须先提供用户名和密码进行验证。它是 samba 的默认配置
中	server	与 user 安全级别类似,但用户名和密码是递交到另外一个服务器去验证,如果验证失败,就退回到 user 安全级
高	domain	要求网络上存在一台 Windows 的主域控制器, Samba 把用户名和密码递交给它去验证

基于以上分析可知,如果图 5-7 所示的网络允许任何用户都可以匿名方式访问 Samba 服务器上的共享资源,那么 smb.conf 文件中应添加一条 security=share 的配置语句, Samba 的安全级别应选择最低的共享方式(share)。

【问题 5】(2 分)

本配置文件中,path 语句用于指定共享的路径是/home/WG_Share。另外它可以与

Samba 变量配合使用。

`writable= yes` 语句用于指定该共享目录缺省是允许写入的。

`valid users` 语句用于指定能够使用该共享资源的用户和组。其中, Samba 变量 `%U` 表示当前会话的用户名。

表 5-11 是对几种常见 Samba 变量定义的归纳(注意:该变量要区分大小写字母)。

表 5-11 几种常见 Samba 变量

变量	定 义	变量	定 义
<code>%a</code>	客户机的结构	<code>%N</code>	NIS 服务器名
<code>%d</code>	当前服务进程的 ID	<code>%P</code>	当前服务的根目录
<code>%G</code>	当前对话的用户的主工作组	<code>%p</code>	NIS 服务的 Home 目录
<code>%g</code>	当前用户所在的主工作组	<code>%R</code>	采用的协议等级
<code>%H</code>	当前服务的用户的 Home 目录	<code>%S</code>	当前服务名
<code>%h</code>	运行 Samba 服务机器的主机名	<code>%T</code>	当前日期和时间
<code>%I</code>	客户机的 IP	<code>%U</code>	当前会话的用户名
<code>%L</code>	服务器的 NETBIOS 名称	<code>%u</code>	当前服务的用户名
<code>%M</code>	客户机的主机名	<code>%v</code>	Samba 服务的版本号
<code>%m</code>	客户机的 NETBIOS 名称		

【问题 6】(2 分)

对 `smb.conf` 文件的配置信息进行修改后,可执行 `testparm` 命令来测试该文件语法设置是否正确。测试通过后一定要执行 `service smb restart` 命令来重启 `smb` 服务。

在缺省情况下,Red Hat Linux 的 Samba 在启动时就已投入运行。也可利用系统脚本来手工启动和停止 Samba。

```
# /etc/rc.d/init.d/smb start (启动 Samba)
```

```
# /etc/rc.d/init.d/smb stop (停止 Samba)
```

【问题 7】(2 分)

编译 Samba 时,需先解压 `samba-3.0.0.tar.gz` 压缩包,然后设置环境变量:

```
$ export LD_LIBRARY_PATH=/usr/local/hybus-arm-linux-R1.1/lib/
```

```
$ export CPPFLAGS=-I/usr/local/hybus-arm-linux-R1.1/include
```

```
$ export LDFLAGS=-L/usr/local/hybus-arm-linux-R1.1/lib/
```

```
$ export CC=/usr/local/hybus-arm-linux-R1.1/arm-linux-gcc
```

```
$ export CFLAGS="-march=armv4 -mtune=arm9tdmi -mshort-load-bytes"
```

接着运行 `configure` 命令:

```
$ ./configure --host=i686 --target=arm-linux --enable-cups=no
```

并修改 `include/config.h` 文件,添加以下一行配置语句:

```
#define HAVE_GETTIMEOFDAY_TZ 1
```

最后运行 `make` 命令生成可执行文件。

如果准备将 Samba 应用到开发板,则需先在准备烧写到 flash 的文件系统中创建以下目录:

```
/usr/local/samba
/usr/local/samba/bin
/usr/local/samba/var
/usr/local/samba/private
/usr/local/samba/lib
```

将编译后的 Samba 源码目录下的 bin/子目录里的内容拷贝到 /usr/local/samba/bin。接着配置 smb.conf 文件,添加、删除及修改 Samba 提供的多种服务。然后用 smbpasswd-a username 命令添加一个 Samba 用户,并生成 Samba 密码文件 smbpasswd,接着运行 Samba 守护进程。

为了方便读者深入理解,灵活掌握 smb.conf 文件的配置,以下是对试题中给出的 smb.conf 配置文件的完整解释:

定义该 Samba 服务器所在的工作组。

```
workgroup=WG_share
```

对 Samba 服务器进行描述,可与 Samba 设定的变量配合使用。%h=运行 Samba 服务机器的主机名。

```
server string=%h
```

设置允许访问共享网络的网段地址和主机 IP 地址。

```
hosts allow=172.16.100. 172.16.200.3 127.0.0.1
```

指定日志文件的存储位置(一般存储在 /var/log/samba/%m.log 文件中)。

```
log file=/var/log/samba/%m.log
```

设置日志文件的最大容量(单位:KB),当 size=0 时,表示对日志文件的大小没有限制。

```
max log size=0
```

设置共享目录磁盘空间的大小(单位:MB),当 size=0 时,表示对磁盘空间的大小没有限制。

```
max disk size=0
```

设置同一客户端能打开的最大文件数。

```
max open file=8
```

设置 Samba 的安全级别,允许任何用户都可以匿名方式访问 Samba 服务器上的共享资源。

```
security=share
```

指定密码里允许的大写字母个数,当参数值为 0 时,Samba 只做两次密码验证:一次是所接收到的密码,另一次是将接收到的密码全部变为小写后的密码。

```
password level=0
```

指定用户名里允许的大写字母个数。

```
username level=0
```

设置是否对密码进行加密。

```
encrypt passwords=yes
# 指定存放 Samba 用户密码的文件。
smb passwd file=/etc/samba/smbpasswd
# 指定存放用户名的映射文件。
username map=/etc/samba/smbusers
# 设置 socket 的编程参数, 以实现最好的文件传输性能。
socket options=TCP_NODELAY SO_RCVBUF=8192 SO_SNDBUF=8192
# 不启用域浏览代理服务功能。
dns proxy=no
[WG_Share] //指定网络邻居里面所看见的共享文件夹的名字
# 对共享文件夹进行注释。
comment=MY Share
# 指定共享的路径为/home/WG_Share。
path=/home/WG_Share
# 指定该共享目录缺省是否可写。
writable=yes
# 指定能够使用该共享资源的用户和组。%U=当前对话的用户名。
valid users=%U
# 指明新建的文件的属性, 一般是 0755。
create mode=0755
# 指明新建的目录的属性, 一般是 0755。
directory mode=0755
[printers] //共享打印机设置段
# 指定的组成员所打印的文件缓冲的目录
path=/var/spool/samba
# 指定该共享缓冲目录不允许写操作
writeable=no
# 设置来宾用户能使用共享打印机
guest ok=yes
# 指明该打印机可以被共享
printable=yes
# 指明该共享打印机的型号
printer driver=HP color LaserJet 8550
```

三、参考答案

表 5-12 给出了本案例问题 1 至问题 7 的参考答案, 供读者练习时参考, 以便查漏补缺。读者也可依照所给出的评分标准得出自己的测试分数, 从而大致评估自己对这一方面知识点的掌握程度。

表 5 - 12 参考答案及评分标准

问题与分值	参考答案及评分标准	自评分
【问题 1】 (3 分)	(1) NETBIOS 协议或 SMB 协议(1 分) (2) SMB 协议或 NETBIOS 协议(1 分) (3) redhat-config-samba(1 分)	
【问题 2】 (2 分)	(4) 172.16.100. 172.16.200.3 127.0.0.1 或 172.16.100. 172.16.200.3 172.16.1.1 (2 分)	
【问题 3】 (2 分)	(5) 0(1 分) (6) 0(1 分)	
【问题 4】 (2 分)	(7) share(1 分) Samba 用户密码存放在/etc/samba/smbpasswd 文件中(1 分)	
【问题 5】 (2 分)	用于指定共享的路径为/home/WG_Share, 而且该共享目录缺省为允许写入(1 分)。设置当前对话的用户为能够使用该共享资源的用户和组(1 分)	
【问题 6】 (2 分)	(8) testparm(1 分) (9) service smb restart(1 分)	
【问题 7】 (2 分)	(10) /usr/local/samba/bin(1 分) (11) 一个 Samba 用户, 并生成 Samba 密码文件 smbpasswd(1 分)	

5.2.5 案例 5 TCP / IP 网络配置

一、案例描述

阅读以下利用 Linux 主机实现 TCP/IP 网络互联的技术说明, 请将以下(1)~(15)空缺处填写完整。(15 分)

【说明】

某实验室 4 台 Linux 主机通过图 5 - 8 所示的方式互连。请将以下(1)~(15)空缺处填写完整, 以实现主机 PC1 与主机 PC4 之间的相互访问。

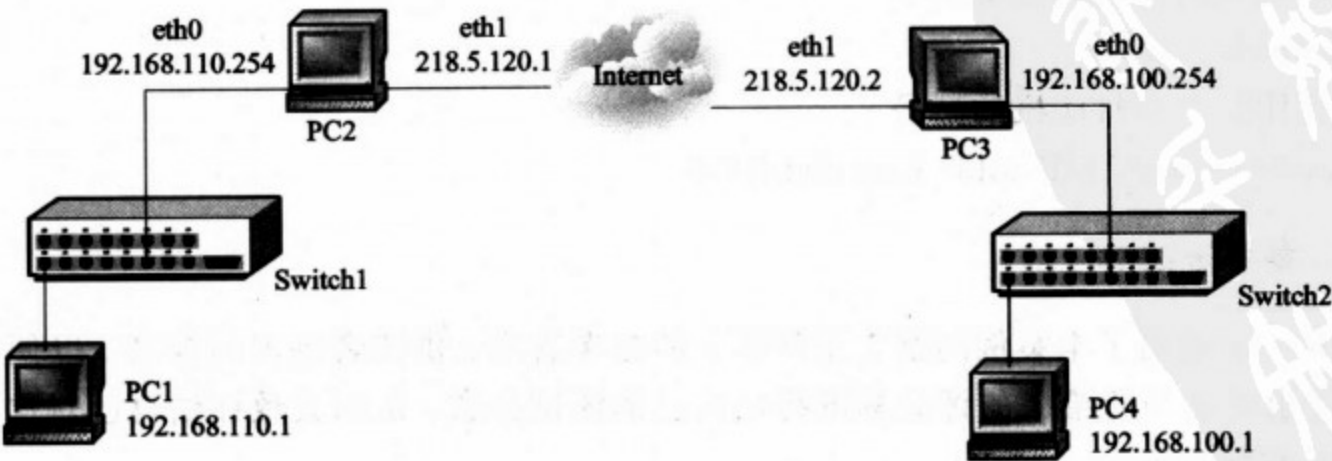


图 5 - 8 4 台 Linux 主机网络互联拓扑图

【问题 1】(4 分)

1. 运行(1)命令关闭主机 PC2 和主机 PC3, 分别在这两台主机上添加第 2 块网卡(eth1)。
2. 在 Linux 操作系统下安装网卡, 如果操作系统没有内置的驱动程序, 那么用户必须(2), 才能完成驱动程序的安装。
3. 在主机 PC2 与 PC3 上为第 2 块网卡分配 IP 地址, 并激活该网络接口。对于主机 PC2 应执行的配置命令是(3); 对于主机 PC3 应执行的配置命令是(4)。

【供选择的答案】

- (1) A. reboot B. shutdown
C. init 1 D. init 6
- (2) A. 用 ifconfig 命令配置网卡 B. 手工修改“/dev/eth0”文件
C. 手工安装驱动程序且重新编译 Linux 内核 D. 重启操作系统
- (4) A. ifconfig -up eth1 218.5.120.2/24
B. ifconfig eth1 218.5.120.2 255.255.255.0 up
C. ifconfig eth1 218.5.120.2 up netmask 255.255.255.0
D. ifconfig eth1 218.5.120.2/8; ifconfig eth1 up

【问题 2】(6 分)

1. 如果使用 routed 作为路由器进程, 则作为路由器的主机 PC2 与 PC3 仅能支持的路由协议是(5); 如果要在主机 PC2 与 PC3 上设置静态路由信息并开启路由功能, 那么对于主机 PC2, 则应执行的配置命令是(6); 对于主机 PC3, 则应执行的配置命令是(7);
2. 在主机 PC1 和 PC4 上配置各自的默认网关, 对于主机 PC1 应执行的配置命令是(8); 对于主机 PC4, 则应执行的配置命令是(9)。通常在主机 PC1 上测试与网关是否连通的最简单命令是(10)。

【供选择的答案】

- (5) A. RIP B. BGP
C. OSPF D. EGP
- (6) A. route add -net 192.168.100.0/24 gw 218.5.120.1
B. route add -net 192.168.100.0/24 gw 218.5.120.2
C. route add-net 192.168.100.0/24 gw 218.5.120.2
echo "1">/proc/sys/net/ipv4/ip_forward
D. route add-net 192.168.110.0/24 gw 218.5.120.1;
echo "1">/proc/sys/net/ipv4/ip_forward
- (8) A. route add default 192.168.110.254
B. route add default 192.168.100.254
C. routeadd-host 192.168.100.1/24 gw 192.168.110.254
D. routeadd-net 0.0.0.0/32 gw 192.168.100.254

【问题 3】(3 分)

在 Linux 网络配置中, 可以通过运行(11)命令来设置主机名字。在不使用 DNS 和 NIS 进行地址解析时, 为保证解析器能找到主机的 IP 地址, 必须将所使用的主机名字写入(12)

文件中。Linux 中提供名字服务的程序是(13)。

【供选择的答案】

- (11) A. route B. ifconfig C. host D. hostname
 (12) A. /etc/networks B. /etc/hosts C. /etc/configs D. /etc/address
 (13) A. named B. address C. nat D. resolvcr

【问题 4】(2 分)

主机 PC1 和 PC4 的网络参数配置后,如果两台主机不能相互共享资源,用(14)命令来测试数据包是否能够到达网关计算机。如果数据包可以到达网关但是不能转发到目标计算机上,则需要用命令 `cat /proc/sys/net/ipv4/ip_forward` 来确认网关计算机的内核是否支持 IP 转发。如果不支持,则该命令输出(15)。

【供选择的答案】

- (14) A. traceroute B. tracert C. nslookup D. route
 (15) A. 1 B. 0 C. yes D. no

二、要点解析

【问题 1】(4 分)

这是一道要求读者掌握 Linux 环境下 TCP/IP 网络安装与配置方面的推理填空题。

通过阅读图 5-8 所示的网络拓扑结构,可先将图中各主机网络接口的 IP 地址信息整理出来,见表 5-13。

表 5-13 各主机网络接口 IP 地址信息

主机	网卡	IP 地址	所在网段	默认网关	子网掩码
PC1	eth0	192.168.110.1	192.168.110.0/24	192.168.110.254	255.255.255.0
PC2	eth0	192.168.110.254			
	eth1	218.5.120.2	218.5.120.0/24		
PC3	eth0	192.168.100.254	192.168.100.0/24		
	eth1	218.5.120.1	218.5.120.0/24		
PC4	eth0	192.168.100.1	192.168.100.0/24	192.168.100.254	

Linux 系统中 TCP/IP 的配置包括主机名的设置、网络接口的 IP 地址配置以及路由的配置。其中:

① 主机名的设置可用 `hostname` 命令来完成,如 `hostname my_pc.gczone.com` 就可以设置机器的主机名。

② 网卡接口的 IP 地址配置可用 `ifconfig` 命令来完成,如要将 IP 地址 192.168.100.254 分配给接口(或网卡)eth0,将 IP 地址 218.5.120.2 分配给接口(或网卡)eth1,则可通过以下两条命令完成这些分配工作。

```
ifconfig eth0 192.168.100.254 netmask 255.255.255.0
```

```
ifconfig eth1 218.5.120.2 netmask 255.255.255.0
```

以上命令中 `netmask` 是设置接口(或网卡)的子网掩码,后面紧跟的是掩码值。

③ 路由的配置可用 `route` 命令来完成,如要将 IP 地址 218.5.120.2/24 设为 218.5.120.0 网段的缺省网关,则可输入“`route add default gw 218.5.120.2`”命令来完成这一设置。

当输入“`route add -net 192.168.100.0 eth0`”命令时,说明通过该路由器的网络接口 `eth0` 能访问的网络是 192.168.100.0,即与本机 `eth0` 接口直连的网络。

在 Linux 环境下有 `shutdown`、`halt`、`reboot` 及 `init` 等一些常用的关机/重启命令,它们都可以达到重启系统的目的,但每个命令的内部工作过程是不同的。其中,在系统关机前使用 `shutdown` 命令,系统就会将这一关机信息通知所有登录的用户,并将 `login` 指令冻结(即新的用户不能再登录)。本案例中,由于要为主机 PC2、PC3 添加第 2 块网卡(`eth1`),根据计算机使用常识可知,这一操作需在关闭计算机电源的状态下完成,因此(1)空缺处可使用能将系统安全关机的 `shutdown` 命令来关闭计算机 PC2、PC3。对于其他供选择的答案,`reboot` 及 `init 6` 是将操作系统重启的命令;`init 1` 命令用于将操作系统切换到单用户模式。

网卡硬件安装后,如果 Linux 操作系统没有内置的驱动程序,那么用户必须手工安装驱动程序且重新编译 Linux 内核,才能完成驱动程序的安装。

由于(3)空缺处是填空题,而(4)空缺处是选择题,因此可以先完成(4)空缺处的选择,从中建立解题的思路,再回头完成(3)空缺处的填写。从图 5-8 网络拓扑图中可知,主机 PC3 上 `eth1` 接口的 IP 地址是 218.5.120.2,该地址属于 C 类 IP 地址,其默认的子网掩码是 255.255.255.0。因此可通过 `ifconfig` 命令为主机 PC3 的 `eth1` 接口分配 IP 地址,并激活该网络接口。其配置命令是:`ifconfig eth1 218.5.120.2 up netmask 255.255.255.0`,其中参数 `up` 表示启动(激活)网卡。如不使用 `netmask` 选项来指定子网掩码,系统会自动加上掩码 255.255.255.0,即命令 `ifconfig eth1 218.5.120.2 up` 与以上配置命令等效。

同理,图 5-8 中主机 PC2 上 `eth1` 接口的 IP 地址是 218.5.120.1,子网掩码是 255.255.255.0。参照(4)空缺处的语法格式,(3)空缺处可填入命令 `ifconfig eth1 218.5.120.1 up netmask 255.255.255.0`,以完成对主机 PC2 的 `eth1` 接口分配 IP 地址,并激活该网络接口。

【问题 2】(6 分)

目前,Linux 系统中路由器进程 `routed` 只支持一个简单的内部网关路由协议——RIP 协议。如果使用 `routed` 作为路由器进程,则作为路由器的主机 PC2、PC3 仅能支持 RIP 动态路由信息协议。因此(5)空缺处应选择 A 选项。对于其他供选择的答案,BGP 是指边界网关协议,OSPF 是指开放最短路径优先路由协议,EGP 是指外部网关协议。

在主机 PC2、PC3 上可使用 `route` 命令来设置静态路由信息并开启路由功能。为分辨网络地址是否划分了子网,使用参数 `add` 来增加一个路由表项;使用 `-net` 选项通知 `route` 进程设置访问网络的网络接口,并开启路由功能。参数 `gw` 可设置本路由器到达目标网络的下一跳地址。因此(6)空缺处填写的主机 PC2 声明到达目标网段 192.168.100.0/24 的路由配置命令,其下一跳 IP 地址就是主机 PC3 的 `eth1` 接口的 IP 地址 218.5.120.2。主机 PC2 上执行的路由配置命令是

```
route add -net 192.168.100.0/24 gw 218.5.120.2
echo "1">/proc/sys/net/ipv4/ip_forward
```

参照(6)空缺处的语法格式,结合图 5-8 所示的拓扑信息,(7)空缺处可填入主机 PC3 上执行的路由配置命令:

```
route add -net 192.168.110.0/24 gw 218.5.120.1
echo "1">/proc/sys/net/ipv4/ip_forward
```

为了能够使用网关服务,必须为网络层提供附加的路由信息,因此这一配置任务也可通过 route 命令来完成。由图 5-8 所示的拓扑结构可知,对于主机 PC1,它通过自己所在网段的路由器(主机 PC2)eth0 接口访问网络上的所有主机,因此需将主机 PC2 的 eth0 接口 IP 地址 192.168.110.254 设置为默认网关。(8)空缺处应选择在主机 PC1 上执行的默认网关设置命令 route add default 192.168.110.254。

结合图 5-8 所示的拓扑信息,参照(6)空缺处的语法格式,可推理出(9)空缺处填入的主机 PC4 上执行的默认网关配置命令 route add default 192.168.100.254。在主机 PC1 上可使用 ping 命令来测试它与网关 192.168.110.254 是否连通,其不带参数的命令格式是 ping 192.168.110.254。

【问题 3】(3 分)

在 Linux 网络配置中,可以通过运行 hostname 命令来设置主机名字;在不使用 DNS 和 NIS 进行地址解析时,为保证解析器能找到主机的 IP 地址,必须将所使用的主机名字写入/etc/hosts 文件中;解析器的功能是实现主机名字与 IP 地址的互查;Linux 中提供名字服务的程序是 named;配置文件“host.conf”的主要作用是规定解析器所使用的服务及顺序。

【问题 4】(2 分)

在 Linux 操作系统中,使用基于 UDP 协议的 traceroute 命令进行路由跟踪,该命令在 Windows 操作系统中对应的是“tracert”。计算机可以通过 nslookup 命令测试 DNS 的配置情况,或实现某个域名及其对应的 IP 地址间的相互查询。route 命令主要用于创建或删除某条路由。

标准的 GNU/Linux 提供了很多可调节的内核参数。例如在 /proc 虚拟文件系统中存在一些可调节的内核参数。这个文件系统中的每个文件都表示一个或多个参数,它们可以通过 cat 工具进行读取,或使用 echo 命令进行修改。以下例子展示了如何查询或启用一个可调节的参数的实验过程。

```
[root@camus]# cat /proc/sys/net/ipv4/ip_forward /* 查询计算机的内核是否支持 IP 转发 */
0 /* 查询结果,0 表示不支持或未启用 */
[root@camus]# echo "1" > /poc/sys/net/ipv4/ip_forward /* 在 TCP/IP 栈中启用 IP 转发 */
[root@camus]# cat /proc/sys/net/ipv4/ip_forward /* 继续查询计算机的内核是否支持 IP 转发 */
1 /* 查询结果,1 表示支持或已启用 */
```

三、参考答案

表 5-14 给出了本案例问题 1 至问题 4 的参考答案,供读者练习时参考,以便查漏补缺。读者可按每空 1 分的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。本案例试题所考查的知识点较细致,对读者在掌握知识的深度上提出了较高的要求。

表 5-14 参考答案及评分标准

问题与分值	参考答案及评分标准	自评分
【问题 1】 (4 分)	(1) B 或 shutdown (2) C 或手工安装驱动程序且重新编译 Linux 内核 (3) ifconfig eth1 218.5.120.1 up netmask 255.255.255.0 (4) C 或 ifconfig eth1 218.5.120.2 up netmask 255.255.255.0 (每空 1 分)	
【问题 2】 (6 分)	(5) A 或 RIP (6) C 或 route add-net 192.168.100.0/24 gw 218.5.120.2 echo "1">/proc/sys/net/ipv4/ip_forward (7) route add-net 192.168.110.0/24 gw 218.5.120.1 echo "1">/proc/sys/net/ipv4/ip_forward (8) A 或 route add default 192.168.110.254 (9) route add default 192.168.100.254 (10) ping 192.168.110.254 (每空 1 分)	
【问题 3】 (3 分)	(11) D 或 hostname (12) B 或 /etc/hosts (13) A 或 named (每空 1 分)	
【问题 4】 (2 分)	(14) A 或 traceroute (15) B 或 0 (每空 1 分)	



第6章 考点热身预测试题

6.1 模拟试题1

6.1.1 上午试题

(考试时间 9:00—11:30 共 150 分钟)

请按下述要求正确填写答题卡:

- (1) 在答题卡的指定位置上正确写入你的姓名和准考证号,并用正规 2B 铅笔在你写入的准考证号下填涂准考证号。
- (2) 本试卷的试题中共有 75 个空格,需要全部解答,每个空格 1 分,满分 75 分。
- (3) 每个空格对应一个序号,有 A、B、C、D 四个选项,请选择一个最恰当的选项作为答案,在答题卡相应序号下填涂该选项。
- (4) 解答前请务必阅读例题和答题卡上的例题填涂样式及填涂注意事项。解答时用正规 2B 铅笔正确填涂选项,如需修改,请用橡皮擦干净,否则会导致不能正确评分。

例题

• 2007 年下半年全国计算机技术与软件专业技术资格(水平)考试日期是(88)月(89)日。

- (88) A. 9 B. 10 C. 11 D. 12
(89) A. 1 B. 2 C. 3 D. 4

因考试日期是“11 月 3 日”,故(88)选 C,(89)选 C,应在答题卡序号(88)下对 C 填涂,在序号(89)下对 B 填涂(参看答题卡)。

试题

• 某流水线浮点加法器分为五级,若每一级所需要的时间分别是 6 ns、8 ns、5 ns、7 ns 和 6 ns,则此流水线的最大加速比为(1)。

- (1) A. 2.0 B. 4.0 C. 4.5 D. 5.0

• I/O 控制方式有多种,(2)一般用于高效的大型系统中。

- (2) A. 中断方式 B. PPU 方式 C. 查询方式 D. DMA 方式

• 一系统由 A、B、C 三个子系统串联而成,它们的寿命均服从指数分布,其中 A、B

的 MTBF 分别是 200 h、400 h。现要求系统的 MTBF 在 100 h 以上,则子系统 C 的 MTBF 至少为(3)h。

- (3) A. 50 B. 200 C. 400 D. 700

• 某高可靠性计算机系统由图 6-1 所示的冗余部件构成,若每个部件的千小时可靠度为 0.95,则该系统的千小时可靠度 R 为(4)。

- (4) A. 0.9999 B. 0.9975 C. 0.9475 D. 0.8574

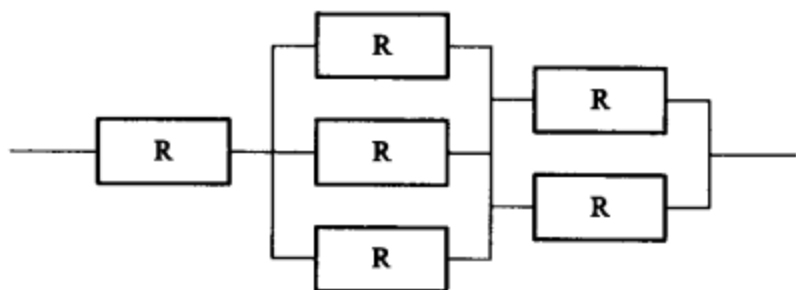


图 6-1 串/并联冗余部件组成图

• 在软件项目管理中可以使用各种图形工具来辅助决策,下面对 Gantt 图的描述不正确的是(5)。

- (5) A. Gantt 图不能表现各个活动的顺序和它们之间的因果关系
B. Gantt 图也不能表现哪些活动具有并行性
C. Gantt 图可以表现各个活动的起始时间
D. Gantt 图也可以表现各个活动完成的进度

• 某工程计划图(如图 6-2 所示),弧上的标记为作业编码及其需要的完成时间(天),作业 F 最迟开始时间需在第(6)天开始。

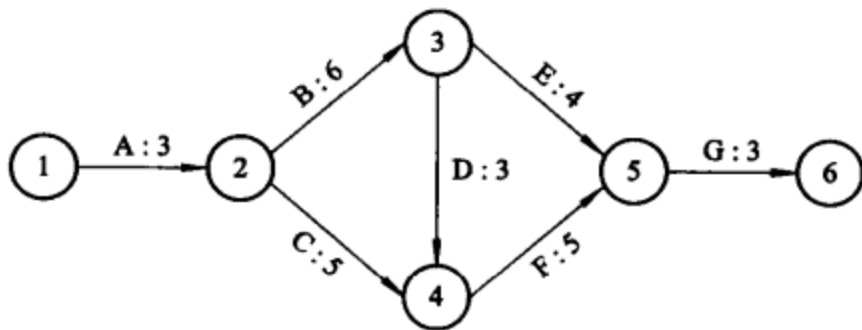


图 6-2 某工程计划图

- (6) A. 8 B. 12 C. 13 D. 17

• ISO 9000 资质认证过程中要对企业的各方面进行严格审查,还要每年进行自检和外检。ISO 9000 质量管理体系认证书的有效期为(7)。

- (7) A. 1 年 B. 3 年 C. 5 年 D. 10 年

• 同事张三、小李为本企业合作开发了一套库存管理信息系统,该系统已通过验收并投入使用。3年后,小李辞职,在 Internet 网上新申请了一个个人网站,为丰富网站内容并宣扬个人工作业绩,小李将该管理软件上传至个人网站的网友下载区中。小李的这种行

- 为(8)。
(8) A. 不构成侵权 B. 至少侵犯该企业的商业秘密权
C. 至少侵犯信息网络传播权 D. 至少侵犯张三的专利权

• 张三从安全认证中心(CA 机构)得到了李四的数字证书(X. 509 格式), 张三可以从该数字证书中得到李四的(9)。

- (9) A. 数字签名 B. 私钥 C. 口令 D. 公钥

• 以下关于防火墙技术的描述, 说法错误的是(10)。

- (10) A. 防火墙可以对请求服务的用户进行控制
B. 防火墙可以对用户如何使用特定服务进行控制
C. 防火墙可以对网络攻击进行反向追踪
D. 防火墙可以对网络服务类型进行控制

• 为了检测系统是否有木马侵入, 可以使用 Windows 操作系统的(11)命令查看当前的活动连接端口。

- (11) A. ipconfig /all B. ping 127. 0. 0. 1
C. netstat - an D. nslookup

• 在某一分页存储管理系统中, 页面的大小为 4 KB。现有一作业, 其部分页表如表 6-1 所示。若给定一个十进制逻辑地址为 8644, 则其十进制物理地址是(12)。

- (12) A. 6186 B. 12 330
C. 33 220 D. 因产生缺页中断, 暂时无法获得

表 6-1 某分页存储管理系统部分页表

页 号	块 号
0	2
1	3
2	8
3	12

• 将带 256 MB 存储卡的数码相机的拍摄分辨率设定为 1600×1200 像素, 颜色深度设定为 24 位, 若不采用压缩存储技术, 则最多可以存储(13)张照片。

- (13) A. 46 B. 47 C. 44 D. 5

• 视频卡中的信号获取器将输入的彩色全电视信号经 A/D 变换和数字解码后在窗口控制器的控制下实时地存入帧存储器中。其中帧存储器的信号为(14)。

- (14) A. 模拟的 RGB 信号 B. 数字的 RGB 信号
C. 模拟的 YUV 信号 D. 数字的 YUV 信号

• CMM 模型将软件过程的成熟度分为五个等级。在(15)级别上, 建立了基本的项目管理过程来跟踪成本、进度和机能, 制定了必要的过程纪律, 并基于以往的项目经验来计划与管理新的项目。

- (15) A. 定义级 B. 管理级 C. 优化级 D. 可重复级

• 软件开发中的瀑布模型典型地刻画了软件生存周期各个阶段的划分, 与其最相适应的软件开发方法是(16)。

- (16) A. 构件化方法 B. 结构化方法
C. 面向对象方法 D. 快速原型方法

• 测试是保证软件质量的重要手段。根据国家标准 GB 8566—88《计算机软件开发规范》的规定,应该在(17)阶段制定系统测试计划。

(17) A. 需求分析 B. 概要设计 C. 详细设计 D. 系统测试

• 已知3个类R、S和T,类R中定义了一个私有方法F1和一个公有方法F2,类S中定义了一个公有方法F3,类S为类R的派生类,类T为类S的派生类,它们的继承方式如下所示:

```
class S: public R {...};
```

```
class T: private S {...};
```

在关于类S的描述中正确的是(18),在关于类T的描述中正确的是(19)。

(18) A. 类S的对象可以访问F1,但不能访问F2
B. 类S的对象可以访问F2,但不能访问F1
C. 类S的对象既可以访问F1,也可以访问F2
D. 类S的对象既不能访问F1,也不能访问F2

(19) A. 类T的对象可以访问F1、F2和F3
B. 类T的对象可以访问F2和F3,但不能访问F1
C. 类T的成员可以访问F2和F3,但不能访问F1
D. 类T的成员不能访问F1、F2和F3

• 16个微处理器的编号分别为0、1、2、3、…、14、15,采用四维立方体单级互联函数为Cube3时,5号微处理器与(20)号微处理器相连接。

(20) A. 10 B. 11 C. 12 D. 13

• 逻辑函数 $F=A\oplus B$ 和 $G=A\odot B$ 满足关系(21)。

(21) A. $F=\bar{G}$ B. $F'=\bar{G}$ C. $F=G\oplus 0$ D. $F=\bar{G}\oplus 1$

• 下列函数中,为最小项之和形式的是(22)

(22) A. $Y_1(A, B, C)=A+\bar{B}C$ B. $Y_2(A, B, C)=\bar{A}\bar{B}C+\bar{A}B\bar{C}+\bar{A}B\bar{C}$
C. $Y_3(A, B, C)=ABC+\bar{A}\bar{B}\bar{C}$ D. $Y_4(A, B, C, D)=ACD+\bar{A}\bar{C}D+AB\bar{D}$

• 下列各类门电路中,可以将输出端直接并联实现“线与”逻辑的门电路是(23)。

(23) A. 三态输出的门电路 B. 集电极开路输出的TTL门电路
C. 互补输出结构的CMOS门电路 D. 推拉式输出结构的TTL门电路

• 由n位触发器构成的扭环形计数器,其无关状态数有(24)个。

(24) A. 2^n-n B. $2n$ C. 2^n-2n D. 2^n-1

• 下面关于RISC与CISC计算机的论述中,不正确的是(25)。

(25) A. RISC计算机采用优化的编译程序,有效地支持高级语言
B. RISC计算机尽量少用通用寄存器,把芯片面积留给微程序
C. CISC计算机可以对存储器和寄存器进行算术和逻辑操作
D. CISC计算机编码长度可变,寻址方式种类丰富

• 以下关于知识产权核(IP Core)的说法错误的是(26)。

(26) A. 知识产权核设计是片上系统(SOC)设计的基础
B. IP固核是基于半导体工艺的物理设计,已有固定的拓扑布局和具体工艺,并已经过工艺验证,具有可保证的性能

- C. IP 软核通常是用硬件描述语言文本形式提交给用户, IP 知识产权不易保护
- D. IP 硬核无需提供寄存器转移级文件, 易于实现 IP 保护, 但其灵活性和可移植性较差

• 对于 16 位嵌入式微处理器体系结构, 半字的位长度是(27)。

- (27) A. 32 位 B. 16 位 C. 12 位 D. 8 位

• 位于主存和嵌入式微处理器内核之间, 存放最近一段时间微处理器使用最多的程序代码和数据的存储器是(28)。

- (28) A. MMC B. Cache C. NOR Flash D. SDRAM

• 嵌入式系统硬件初始化过程分为三个主要环节, 按照自下而上、从硬件到软件的次序依次是(29)。

- (29) A. 板级→系统级→片级 B. 系统级→板级→片级
C. 片级→板级→系统级 D. 系统级→片级→板级

• 关于计算机性能的评价, 下列说法中错误的是(30)。

- (30) A. 微处理器主频高的机器比主频低的机器速度快
B. 平均指令执行速度(MIPS)能正确反映计算机执行实际程序的速度
C. 基准程序测试法能比较全面地反映实际运行情况, 但各个基准程序测试的重点不一样
D. EEMBC 测试基于每秒钟算法执行的次数和编译代码大小的统计结果

• 通常嵌入式文件系统不具有(31)功能。

- (31) A. 提供建立、修改、改变和删除目录等服务
B. 提供创建、打开、读写、关闭和撤销文件等服务
C. 提供对文件的加密和解密功能
D. 设置、修改对文件和目录的存取权限

• 以太网的数据帧结构如图 6-3 所示, 包含在 TCP 段中的数据部分最长应该是(32)字节。

目标MAC地址	源MAC地址	协议类型	IP头	TCP头	数据	CRC
---------	--------	------	-----	------	----	-----

图 6-3 以太网的数据帧结构图

- (32) A. 1434 B. 1460 C. 1480 D. 1500

• 在 TCP/IP 网络中, 为各种公共服务保留的 TCP 端口号范围是(33)。

- (33) A. 1~255 B. 1~1023 C. 1~1024 D. 1~65 535

• IEEE 802.11 标准使用的传输技术主要有(34)。

- (34) A. 红外线、跳频扩频与蓝牙 B. 红外线、跳频扩频与直接序列扩频
C. 跳频扩频、直接序列扩频与蓝牙 D. 红外线、直接序列扩频与蓝牙

• 以下关于嵌入式实时系统说法错误的是(35)。

- (35) A. 实时系统要求在任何情况下, 结果产生的时间都应该是可以预期的
B. 实时操作系统采用虚拟内存管理方式实现对内存资源的合理分配和存取
C. 实时操作系统中, 为了能够保证响应时间, 应当允许具有较高优先级的任务

能够抢占低优先级的任务运行

D. 由于文件系统对于实时系统来说不是必需的, 因此实时操作系统中的文件系统部分最好是一个可以自由裁减的组件

• 若在一个 32 位字长的嵌入式微处理器上定义了 1 个 int 类型的常量 $a = 0x8192F3A5$, 按大端存储法存放于 $0x6000$ 内存地址处, 以下选项中能正确表达这种存放顺序的是(36)。

(36)

地址 选项	6000	6001	6002	6003
A	81	92	F3	A5
B	18	29	3F	5A
C	A5	F3	92	81
D	5A	3F	29	18

• 根据程序局部性理论, Denning 提出了工作集理论。工作集是进程运行时被频繁访问的页面集合。在进程运行时, 如果它的工作页面都在(37)器内, 则能够使用进程有效地运行, 否则会出现频繁的页面调入/调出现象。假设窗口尺寸为 10, 在某一段时间内, 任务所访问的逻辑页面顺序如图 6-4 所示, 那么在 t_1 时刻的工作集是(38)。

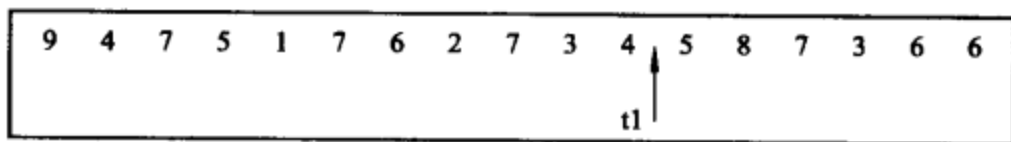


图 6-4 某任务访问逻辑页面的顺序图

(37) A. 外部存储 B. 主存储 C. 辅助存储 D. 虚拟存储

(38) A. {1, 2, 3, 4, 5, 6, 7} B. {1, 2, 3, 4, 4, 5, 6, 7, 7, 7}

C. {1, 2, 3, 4, 5, 6, 7, 9} D. {4, 7, 5, 1, 7, 6, 2, 7, 3, 4}

• 某嵌入式系统的文件分配表如图 6-5 所示, 对于文件 1 总共有(39)个物理块。

文件目录项	文件1					0002		
块号	0	1	2	3	4	5	6	7
FAT表	0001	0002	0005	0004	FFFF	0006	0007	FFFF

图 6-5 某嵌入式系统的文件分配表

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

• 在内部排序中, 通常要对被排序数据序列进行多趟扫描。各种排序方法有其不同的排序实施过程和(时间)复杂性。对给定的整数序列(541, 132, 984, 746, 518, 181, 946, 314, 205, 827)进行从小到大的排序时, 采用快速排序(以中间元素 518 为基准)的第一趟扫描结果是(40)。设被排序数据序列有 n 个元素, 快速排序的复杂性是(41)。

- (40) A. (541, 132, 827, 746, 518, 181, 946, 314, 205, 984)
B. (205, 132, 314, 181, 518, 746, 946, 984, 541, 827)
C. (541, 132, 984, 746, 827, 181, 946, 314, 205, 518)
D. (132, 541, 746, 518, 181, 946, 314, 205, 827, 984)

- (41) A. $O(n \lg n)$ B. $O(n^2)$ C. $O(\lg n)^2$ D. $O(n^2 \lg n)$

• 以下无助于抑制干扰源的措施是(42)。

- (42) A. 通过在干扰源回路串联电感或电阻以及增加续流二极管来减小干扰源的 du/dt
B. 在继电器接点两端并接一个 RC 串联抑制电路
C. 在可控硅两端并接 RC 抑制电路
D. 布线时避免 $\leq 90^\circ$ 的折线

• 以下是部分 ARM 汇编程序, 该程序的执行结果是(43)。

```
AREA EXAMPLE2, CODE, READONLY
ENTRYR
start  MOV ro, #10
      MOV r1, #3
      BL DOADD
doadd  ADD r0, r0, r1
      MOV pc, lr
      END
```

- (43) A. 结果值 13 存放在 R0 寄存器中 B. 结果值 23 存放在 R1 寄存器中
C. 结果值 13 存放在 LR 寄存器中 D. 结果值 23 存放在 LR 寄存器中

• 在 ARM 的通用寄存器 R0~R15 中, 被用做栈指针的寄存器是(44)。

- (44) A. R0 B. R13 C. R14 D. R15

• ARM 是典型的 RISC 体系, 其模式控制位 M[4:0]为 0b10010 时, 代表当前嵌入式微处理器处于(45)模式。

- (45) A. IRQ B. Abort C. Supervisor D. System

• 以下关于禁止存储管理单元(MMU)存储访问过程说法错误的是(46)

- (46) A. 所有物理地址和虚拟地址相等
B. 使用平板存储模式
C. 存储访问不考虑 C 和 B 控制位
D. 存储访问不进行权限控制, MMU 也不会产生存储访问中止信号

• 嵌入式系统支持的内存块为极小页时, 可分成大小为(47)的子页。

- (47) A. 1 KB B. 4 KB C. 16 KB D. 不能再细分

• 在 ARM 体系结构的嵌入式系统中, 快速上下文切换技术控制位编码为 0b10, 则表示的方位类型是(48)。

- (48) A. 没有访问权限 B. 客户类型
C. 保留 D. 管理者权限

• 进行存储解决方案选择时, 以下说法错误的是(49)。

- (49) A. NAND Flash 的读取速度比 NOR Flash 快一些
B. NAND Flash 的写入速度比 NOR Flash 快一些

C. NAND Flash 的擦除速度比 NOR Flash 快一些

D. NAND Flash 的写入操作需要先进行擦除操作

- 关于 SRAM 的写操作周期, 下列说法错误的是(50)。

(50) A. CE 端为低电平

B. R/W' 端为高电平

C. 地址出现在 address 线上

D. 数据出现在 data 线上

- 交叉编译器软件属于嵌入式系统的(51)。

(51) A. 系统软件

B. 应用软件

C. 支撑软件

D. 中间件

- 设有以下 C 语言说明语句:

```
struct ex { int x ; float y; char z ; } example;
```

则下面的叙述中不正确的是(52)。

(52) A. struct 是结构体类型的关键字 B. x、y、z 都是结构体成员名

C. struct ex 是结构体类型

D. example 是结构体类型名

- 嵌入式 Linux 操作系统中任务的创建过程如下, 以下说法正确的是(53)。

```
void main( )
```

```
{ int pid;
```

```
  pid=fork( );
```

```
  if(pid>0)
```

```
    printf("parent task");
```

```
  else if(pid==0)
```

```
    {printf("child task");
```

```
    execvp("MyTash", NULL);
```

```
  }
```

```
}
```

(53) A. 子任务的创建基于 fork/exec 模型

B. 子任务的创建基于 spawn 模型

C. 先为子任务分配内存空间, 再分配相应的数据结构 ,

D. 直接为子任务分配一个全新的地址空间, 然后再将其代码装入运行

- 下列属于低级任务间通信方式的是(54)。

(54) A. 共享内存

B. PV 操作

C. 消息传递

D. 管道通信

- 以下关于嵌入式系统中实模式说法错误的是(55)。

(55) A. 划分了“系统空间”和“用户空间”

B. OS 内核与外围应用程序之间不再有物理边界

C. 运行上下文和栈是独享内核线程

D. 系统中“任务”或“进程”全都是内核线程

- 假设页面大小为 1 KB, 对于逻辑地址 0xE9B7 所对应的逻辑页面号为(56)。

(56) A. 0x3A

B. 0x1B7

C. 0xE9

D. 0xB7

- (57)主要用于 Linux 系统中进程间相互传递数据。

(57) A. FIFO 文件

B. 设备文件

C. 链接文件

D. 目录文件

- Linux 系统对普通文件默认的访问权限是(58)。

- (58) A. -rwxrwx--- B. -rw-rw-r--
C. -rwx----- D. -rw-r--r--

• 在 Linux 系统中, Samba 服务允许文件和打印机被用户网络中的所有系统共享, 其默认的安全级别是(59)。

- (59) A. share B. user C. server D. domain

• 以下正确描述嵌入式系统的设计流程的是(60)。

- (60) A. 系统需求分析→硬件和软件划分→系统详细设计→迭代与实现→系统集成→系统测试→系统维护
B. 系统需求分析→硬件和软件划分→系统详细设计→系统集成→迭代与实现→系统测试→系统维护
C. 系统需求分析→系统集成→硬件和软件划分→系统详细设计→迭代与实现→系统测试→系统维护
D. 系统需求分析→硬件和软件划分→迭代与实现→系统详细设计→系统集成→系统测试→系统维护

• 模块的耦合度描述了(61)。

- (61) A. 模块内各种元素结合的程度 B. 模块内多个功能之间的接口
C. 模块之间公共数据的数量 D. 模块之间相互关联的程度

• 一个故障已经被发现, 而且也被排除了, 为了检查修改是否引起了其他故障, 这时应该进行(62)。

- (62) A. 程序走查 B. 退化测试 C. 软件评审 D. 接收测试

• 嵌入式应用软件的开发必须将硬件、软件、人力资源等元素集成起来, 并进行适当的组合以实现应用软件对功能和性能的需求。以下正确描述嵌入式应用软件的设计流程的是(63)。

- (63) A. 按照实时性划分软件功能模块→确定硬件驱动的软件接口→生成各模块代码→功能模块的集成测试→调试→代码固化
B. 按照实时性划分软件功能模块→确定硬件驱动的软件接口→生成各模块代码→固化调试→功能模块的集成测试→代码固化
C. 确定硬件驱动的软件接口→按照实时性划分软件功能模块→生成各模块代码→功能模块的集成测试→代码固化→调试
D. 确定硬件驱动的软件接口→按照实时性划分软件功能模块→生成各模块代码→固化调试→功能模块的集成测试→代码固化

• 嵌入式系统需求分析的任务通常不包括(64)。

- (64) A. 确定功能要求 B. 分析数据要求
C. 软硬件模块划分 D. 修正开发计划

• 下面列出了系统维护工作流程中的几个关键步骤, 正确的工作顺序是(65)。

- ① 用户提交维护申请报告 ② 交付使用 ③ 更新文档 ④ 测试
⑤ 核实和评价维护申请报告 ⑥ 制定维护计划 ⑦ 实施维护

- (65) A. ①→⑤→⑥→⑦→④→③→② B. ①→⑥→⑤→⑦→③→④→②
C. ①→⑤→⑥→⑦→③→④→② D. ①→⑥→⑤→⑦→④→③→②

• With hundreds of millions of electronic (66) taking place daily, businesses and organizations have a strong incentive to protect the (67) of the data exchanged in this manner, and to positively ensure the (68) of those involved in the transactions. This has led to an industry-wide quest for better, more secure methods for controlling IT operations, and for deploying strong security mechanisms deeply and broadly throughout networked infrastructures and client devices. One of the more successful concepts to engage the imaginations of the security community has been the development of standards-based security (69) that can be incorporated in the hardware design of client computers. The principle of encapsulating core security capabilities in (70) and integrating security provisions at the deepest levels of the machine operation has significant benefits for both users and those responsible for securing IT operations.

- | | | | |
|--------------------|------------------|-------------------|----------------|
| (66) A. devices | B. transactions | C. communications | D. businesses |
| (67) A. operation | B. completeness | C. integrity | D. interchange |
| (68) A. identities | B. homogeneities | C. creations | D. operations |
| (69) A. appliances | B. chips | C. tools | D. means |
| (70) A. software | B. form | C. computer | D. silicon |

• Pharming is a scamming practice in which malicious code is installed on a personal computer or server, misdirecting users to (71) Web sites without their knowledge or consent. Pharming has been called “phishing without a lure.”

In phishing, the perpetrator sends out legitimate- (72) E-mails, appearing to come from some of the Web’s most popular sites, in an effort to obtain personal and financial information from individual recipients. But in pharming, larger numbers of computer users can be (73) because it is not necessary to target individuals one by one and no conscious action is required on the part of the victim. In one form of pharming attack, code sent in an E-mail modifies local host files on a personal computer. The host files convert URLs into the number strings that the computer uses to access Web sites. A computer with a compromised host file will go to the fake Web site even if a user types in the correct Internet address or clicks on an affected (74) entry. Some spyware removal programs can correct the corruption, but it frequently recurs unless the user changes browsing (75).

- | | |
|--------------------------|-------------------|
| (71) A. few | B. fraudulent |
| C. normal | D. structured |
| (72) A. connecting | B. binding |
| C. looking | D. contenttagging |
| (73) A. victimized | B. personate |
| C. identity | D. control |
| (74) A. hypertext | B. computation |
| C. expectation | D. bookmark |
| (75) A. habits signature | B. site |
| C. tags | D. address |

6.1.2 下午试题

请按下述要求正确填写答题卡：

(1) 在答题卡的指定位置上正确写入你的姓名和准考证号，并用正规 2B 铅笔在你写入的准考证号下填涂准考证号。

(2) 本试卷的试题中共有 75 个空格，需要全部解答，每个空格 1 分，满分 75 分。

(3) 每个空格对应一个序号，有 A、B、C、D 四个选项，请选择一个最恰当的选项作为答案，在答题卡相应序号下填涂该选项。

(4) 解答前请务必阅读例题和答题卡上的例题填涂样式及填涂注意事项。解答时用正规 2B 铅笔正确填涂选项，如需修改，请用橡皮擦干净，否则会导致不能正确评分。

例题

• 2007 年下半年全国计算机技术与软件专业技术资格(水平)考试日期是(88)月(89)日。

(88) A. 9 B. 10 C. 11 D. 12

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

因考试日期是“11 月 3 日”，故(88)选 C，(89)选 C，应在答题卡序号(88)下对 C 填涂，在序号(89)下对 C 填涂(参看答题卡)。

试题一

阅读以下关于嵌入式系统 A/D 接口技术的说明，根据要求回答问题 1 至问题 6。(15 分)

【说明】

12 位逐次逼近式 A/D 转换芯片 AD574A 内置双极性电路的启动转换和读取数据命令均由控制总线提供，具有自动校零和自动极性转换功能，非线性误差小于 $\pm 1/2\text{LSB}$ ，转换时间为 $25\mu\text{s}$ 。AD574A 芯片的内部框图及各引脚的功能图如图 6-6 所示。图 6-7 所示为 AD574A 单极性输入电路原理图。

表 6-2 为 AD574A 工作时控制端标志意义表。

表 6-2 AD574A 工作时控制端标志意义表

CE	$\overline{\text{CS}}$	R/ $\overline{\text{C}}$	12/ $\overline{8}$	A ₀	操 作
0	X	X	X	X	禁止
X	1	X	X	X	禁止
1	0	0	X	0	启动 12 位转换
1	0	0	X	1	启动 8 位转换
1	0	1	接 +5 V	X	12 位并行输出有效
1	0	1	接 0 V	0	高 8 位并行输出有效
1	0	1	接 0 V	1	低 4 位并行输出有效

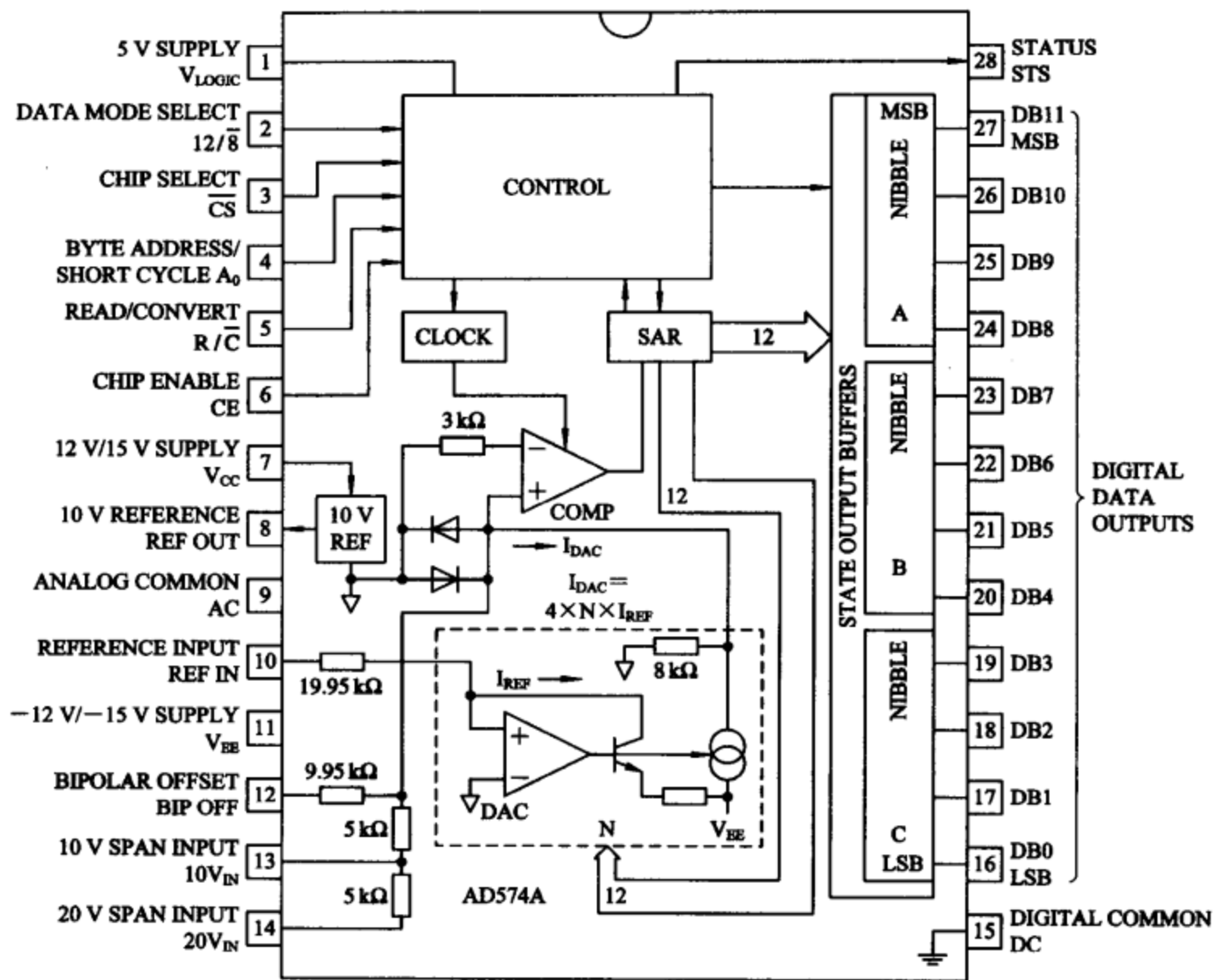


图 6-6 AD574A 芯片的内部框图及各引脚的功能图

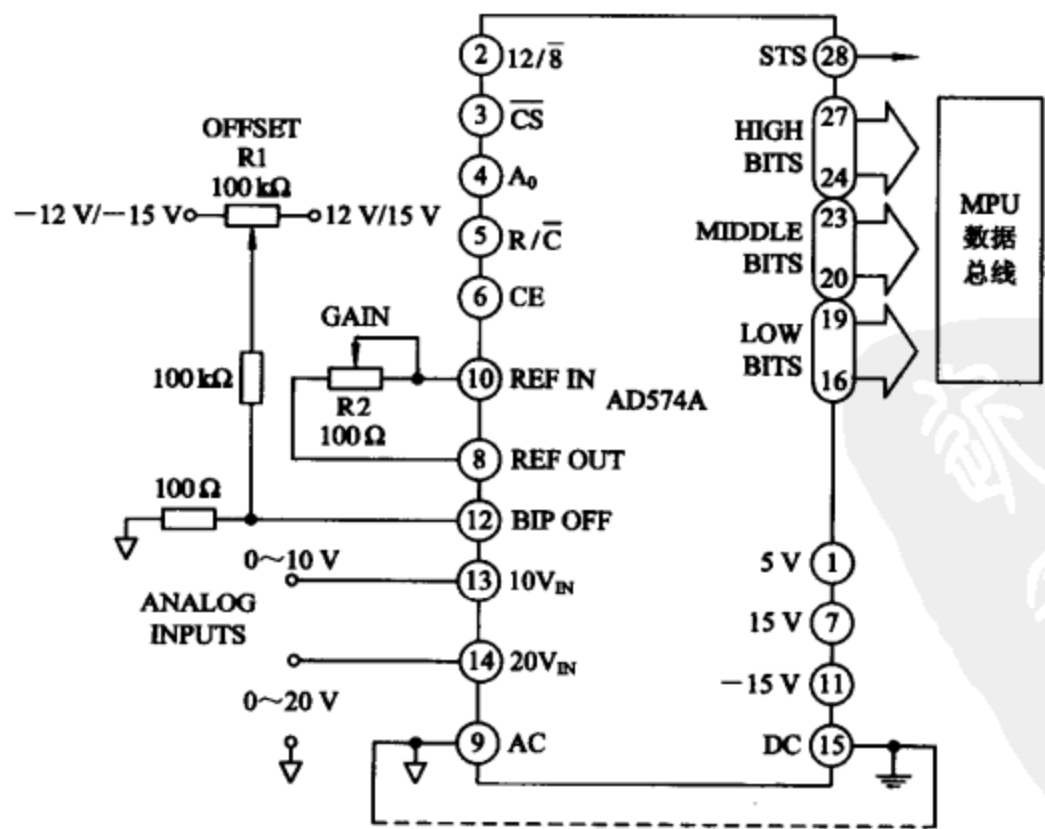


图 6-7 AD574A 单极性输入电路原理图

【问题 1】(2 分)

图 6-7 电路原理图中有哪几种模拟电压输入?

【问题 2】(4 分)

图 6-7 单极性输入电路原理图中, 电位器 R1、R2 分别用于 A/D 转换电路中哪些性能的调整?

【问题 3】(3 分)

AD574A 芯片的工作模式分为全速工作模式和单一工作模式。如果需要 AD574A 工作于单一模式, 只需将(1)和 12/ $\bar{8}$ 端接至 +5 V 电源端, (2)和 A₀ 接至 0 V, 仅用 R/ \bar{C} 端来控制 A/D 转换的启动和数据输出。当 R/ \bar{C} = 0 时, 启动 A/D 转换器, 经 25 μ s 后 STS = 1, 表明 A/D 转换结束, 此时将(3)置 1, 即可从数据端读取数据。

【(1)~(3)空缺处供选择的答案】

A. CE B. A₀ C. R/ \bar{C} D. 12/ $\bar{8}$ E. \bar{CS}

【问题 4】(2 分)

为了测量某材料的性质, 要求以 500 点/s 的速度连续采样 1 min。图 6-7 所示的 A/D 转换电路能否在转换时间指标上满足该要求? 请列出简要计算过程。

【问题 5】(2 分)

若 12 位逐次逼近式 A/D 转换芯片 AD574A 的满量程电压为 10 V, 非线性误差小于 $\pm 1/2$ LBS, 其最小有效位的量化单位为 9.77 mV, 求该 A/D 转换芯片的绝对精度。请列出简要计算过程。

【问题 6】(2 分)

从启动信号到转换结束的数字量输出需要经过一段时间, 然而模拟量转换期间要求模拟信号保持不变, 因此必须使用采样保持电路。请在 150 字以内说明采样保持电路主要完成的功能。

试题二

阅读以下说明和 x86 汇编语言代码, 根据要求回答问题 1 至问题 3。表 6-3 所示为 x86 系统指令及寄存器说明表。(15 分)

【说明】

在计算机控制中, 有些数据补偿、计算及转换等参数的计算非常复杂, 编程难, 程序长且运算费时, 但是用数表却比较容易实现。

设有 256 个字节的数据, 已按从小到大的顺序存放在以 BINTAB 为首地址的数据区单元中, 现要求查找其中值为 126 的数据, 用对分查表法查询的汇编程序如下:

【汇编程序代码】

```
DATA    SEGMENT
BINTAB  DB 18, 25, 32, 46, 56, 78, ...      ; 第 1~100 个数据
          DB 90, 95, 99, 100, 106, 126, ...   ; 第 101~200 个数据
          DB 189, 195, 197, 202, 206, 216, ... ; 第 201~256 个数据
COUNT  EQU    256
FLAG    DW     ?
```

```

KEY      EQU      (1)
DATA     ENDS

STAK     SEGMENT
STPN     DB 20 DUP(?)
STAK     ENDS

CODE     SEGMENT
          ASSUME CS:CODE, DS:DATA, ES:DATA, SS:STAK
BINSEAT PROC FAR
START:   MOV      AX, DATA
          MOV      DS, AX
          MOV      ES, AX
          LEA      SI, BINTAB
          MOV      CX, COUNT
          MOV      DX, 1
          MOV      AX, SI
          ADD      AX, CX
          MOV      DI, AX
          MOV      AL, KEY
LOP0:    MOV      BX, SI
          ADD      BX, DI
          SHR      BX, 1
          CMP      AL, [BX]
          JZ       FOUND
          PUSHF
          (2)
          JZ       NFOUND
          POPF
          JL       LESS
          MOV      SI, BX
          JMP      NEXT
LESS:    (3)
NEXT:    (4)
          JMP      LOP0
NFOUND:  MOV      AX, DX
          ;未找到, 标志全 1 送 DX
FOUND:   MOV      AX, DX
          MOV      FLAG, AX
          RET
BINSEAT ENDP
CODE ENDS
          (5)

```

【问题 1】(10 分)

请根据试题的要求,将汇编程序代码中(1)~(5)空缺处的内容填写完整。

【问题 2】(2 分)

请用 MOV 语句实现汇编程序代码中“LEA SI, BINTAB”语句的功能。

【问题 3】(3 分)

该汇编程序执行后,FLAG 存储单元中的数值是什么?使用对分查表法查询位于这 256 个字节数据表中的数据时,“CMP AL, [BX]”最多被执行几次?

表 6-3 x86 系统指令及寄存器说明

指令	功 能	寄存器		功 能
MOV	传送字或字节	段寄存器	CS	代码段寄存器
LEA	地址传送		DS	数据段寄存器
ADD	加法指令		SS	堆栈段寄存器
SUB	减法指令		ES	附加段寄存器
PUSH	进栈操作	数据寄存器	AX	累加器
POP	出栈操作		BX	基址寄存器
SHR	逻辑右移		CX	计数寄存器
CMP	比较指令		DX	数据寄存器
INC	加 1 指令	专用寄存器	BP	基数指针寄存器
JMP	无条件转移指令		SP	堆栈指针寄存器
JZ/JNZ	结果为 0/不为 0 时转移		SI	源变址寄存器
JL/JG	结果小于/大于转移		DI	目的变址寄存器
JC/JNC	有进(借)位/无进(借)位时转移			
LOOP	CX 不为 0 时循环			
RET	返回主程序			
IRET	从中断服务程序返回			

试题三

阅读以下关于嵌入式系统技术的说明,根据要求回答问题 1 至问题 3。(15 分)

【说明】

某水电站根据安全监控的需要决定采用嵌入式工控微机进行实时多任务安全监测。该系统有 32 路模拟量输入、96 路开关量、16 路脉冲量输入和水位落差量输入,产生显示信息、报警信号、监测报表和水位传感器控制信号等输出。该水电站安全监测系统具有以下几个特点:

(1) 对各种模拟量、开关量、脉冲量、水位落差量等数据量的监测,事实上只是周期性地重复执行相应的各个任务模块。

(2) 监视的对象比较固定,系统硬件的规模相对固定不变。

(3) 尽可能允许把所有软件都固化在 EPROM 中。

(4) 人机交互作用的内容固定,即由操作人员通过键盘去调用此系统中所存入的相应应用程序,不会去修改应用程序。

基于以上安全监测系统的特点,该水电站技术研发部门准备选用现有的工业控制机系统及现成的各种模板等硬件,由本部门的开发人员自行开发软件。所开发的软件主要包括:

(1) 应用程序,即按各类输入/输出量的采集、处理、显示、报警和制表等功能分别作为若干任务块,由这些任务组成应用程序。

(2) 多任务嵌入式实时操作系统,用来协调各任务的资源分配与管理,管理任务之间的联系与多任务操作。

【问题 1】(5 分)

根据以上安全监测系统的特点,技术研发人员郭工程师提出了一个自行编制简化“多任务嵌入式实时操作系统”的方案,以适应于协调管理硬件资源和监测对象的实际需求,其中只保留了标准的多任务实时操作系统的核心功能,例如 CPU 管理、中断管理和外设管理等。请在 200 字以内简要说明郭工程师提出的简化的“多任务嵌入式实时操作系统”可能具有哪些优点。

【问题 2】(5 分)

设计简化的多任务嵌入式实时操作系统时,由于多个任务均可能要求占用 CPU 这个关键资源,因此 CPU 的任务管理是一个非常重要的设计内容。在该嵌入式实时操作系统中,作为占用资源的基本单位,任务总共有五个状态,分别是休眠状态、就绪状态、运行状态、等待状态和延时等待状态。在任何时候,一个任务只会处于其中的某一个状态。请根据图 6-8 中已给出的信息将(1)~(5)空缺处填写完整。

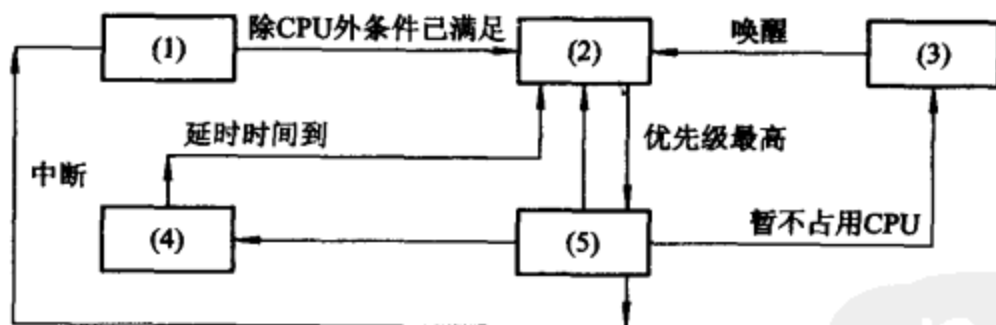


图 6-8 简化的实时操作系统的任务状态转换图

【问题 3】(5 分)

在该水电站安全监测系统中,数据采集任务(Collect_task)把所采集的数据送入一单缓冲区,计算任务(Calculate_task)从该单缓冲区中取出数据进行计算。以下是利用信号量机制实现两个任务共享单缓冲区的 C 语言形式化描述程序。请将程序代码中(6)~(9)空缺处以及(10)空缺处的内容填写完整。

```

int  sf=0;
int  (6);
main( )

```

```
{
    cobegin
        Collect_task();
        Calculate_task();
    coend
}

Collect_task()
{
    While(采集工作未完成)
    {
        采集一个数据;
        P(se);
        将数据送入缓冲区中;
        (7);
    }
}

Calculate_task()
{
    While(计算工作未完成)
    {
        (8);
        从缓冲区中取出数据;
        (9);
        进行数据计算送入缓冲区中;
    }
}
```

如果以上程序中“int sf=0;”语句不小心被改写成“int sf=-1;”,那么系统运行时将会进入(10)状态。

试题四

阅读以下利用场景法设计测试用例的技术说明,回答问题 1 至问题 5(15 分)。

【说明】

现在的软件几乎都是由事件触发来控制流程的,事件触发时的情景便形成了场景,而同一事件不同的触发顺序和处理结果就形成了事件流。这种在软件设计方面的思想也可被引入到软件测试中,生动地描绘出事件触发时的情景,有利于测试设计者设计测试用例,同时测试用例也更容易得到理解和执行。

用例场景是通过描述流经用例的路径来确定的过程,这个流经过程要从用例开始到结束遍历其中所有基本流(基本事件)和备选流(分支事件)。以下表 6-4 是对某 IC 卡加油机应用系统基本流的描述,表 6-5 是对该 IC 卡加油机应用系统备选流的描述。

表 6-4 基本流描述表

序号	用例名称	用例描述
A1	准备加油	客户将 IC 加油卡插入加油机
A2	验证加油卡	加油机从加油卡的磁条中读取账户代码，并检查它是否属于可以接收的加油卡
A3	验证黑名单	加油机验证卡账户是否存在于黑名单中，如果属于黑名单，则加油机吞卡
A4	输入购油量	客户输入需要购买的汽油数量
A5	加油	加油机完成加油操作，从加油卡中扣除相应金额
A6	返回加油卡	退还加油卡

表 6-5 备选流描述表

序号	用例名称	用例描述
B	加油卡无效	在基本流 A2 过程中，该卡不能够识别或是非本机可以使用的 IC 卡，加油机退卡，并退出基本流
C	卡账户属于黑名单	在基本流 A3 过程中，判断该卡账户属于黑名单，例如，已经挂失，加油机吞卡，退出基本流
D	加油卡账面现金不足	系统判断加油卡内现金不足，重新加入基本流 A4，或选择退卡
E	加油机油量不足	系统判断加油机内油量不足，重新加入基本流 A4，或选择退卡

【问题 1】(3 分)

图 6-9 是对该 IC 卡加油机应用系统的基本流路径和备选流路径的描述，请用题干中描述的相应字母将图中(1)~(6)空缺处的内容填写完整。

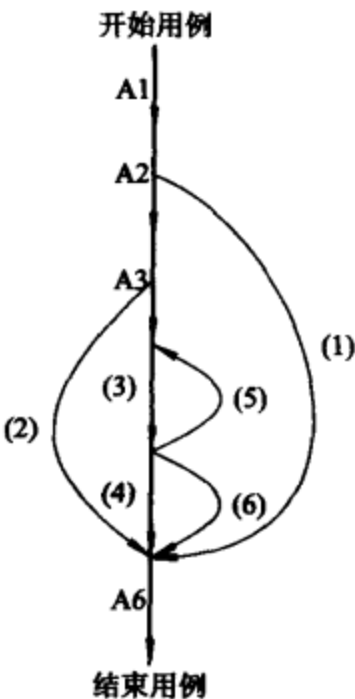


图 6-9 某 IC 卡加油机应用系统场景路径的描述图

【问题 2】(5 分)

场景中的每一个场景都需要确定测试用例，一般采用矩阵或决策表来确定和管理测试用例。表 6-5 是一种通用格式，表中各行代表各个测试用例，而各列代表测试用例的信息。本例中的测试用例包含测试用例 ID 号、场景(或说明/条件)、测试用例中涉及的所有数据元素(作为输入或已经存在于数据库中)以及预期结果等项目。

测试用例的设计步骤通常是：首先确定执行用例场景所需的数据元素(本例中包括账号、是否黑名单卡、输入油量、账面金额以及加油机油量)，然后构建矩阵，最后确定包含执行场景所需的适当条件的测试用例。在表 6-6 的测试矩阵中，V 表示有效数据元素，I 表示无效数据元素，n/a 表示不适用，例如 CW01 表示“成功加油”基本流。请按上述规定为其他应用场景设计测试用例矩阵。

表 6-6 测试用例表

测试用例 ID 号	场景	账号	是否黑名单卡	输入油量	账面金额	加油机油量	预期结果
CW01	场景 1: 成功加油	V	I	V	V	V	成功加油
CW02							
CW03							
CW04							
CW05							

【问题 3】(2 分)

对于基本流 A 来说，表 6-6 所示中哪些测试用例属于正面测试用例，哪些测试用例属于负面测试用例。请用表 6-6 中相应的“测试用例 ID 号”回答问题。

【问题 4】(2 分)

请在 150 字以内简要说明以表 6-6 所示的方法创建测试用例矩阵的优点。

【问题 5】(3 分)

假如加油机内油量足够，油价为 5 元/升，用户的账户金额为 800 元，那么在基本流 A4 输入油量的过程中，请运用边界值分析方法为 A4 选取合适的输入数据(即油量，单位为升)。

试题五

阅读以下关于 CAN 现场总线嵌入式监控系统的技术说明，根据要求回答问题 1 至问题 5。(15 分)

【说明】

某自动化仪表企业正在研究使用现场总线网络构成自动监测系统的可行性，打算采用 CAN(Controller Area Network, 控制器局域网络)总线作为底层网络构件系统。该企业试验用的数据采集与监测系统的网络结构如图 6-10 所示。

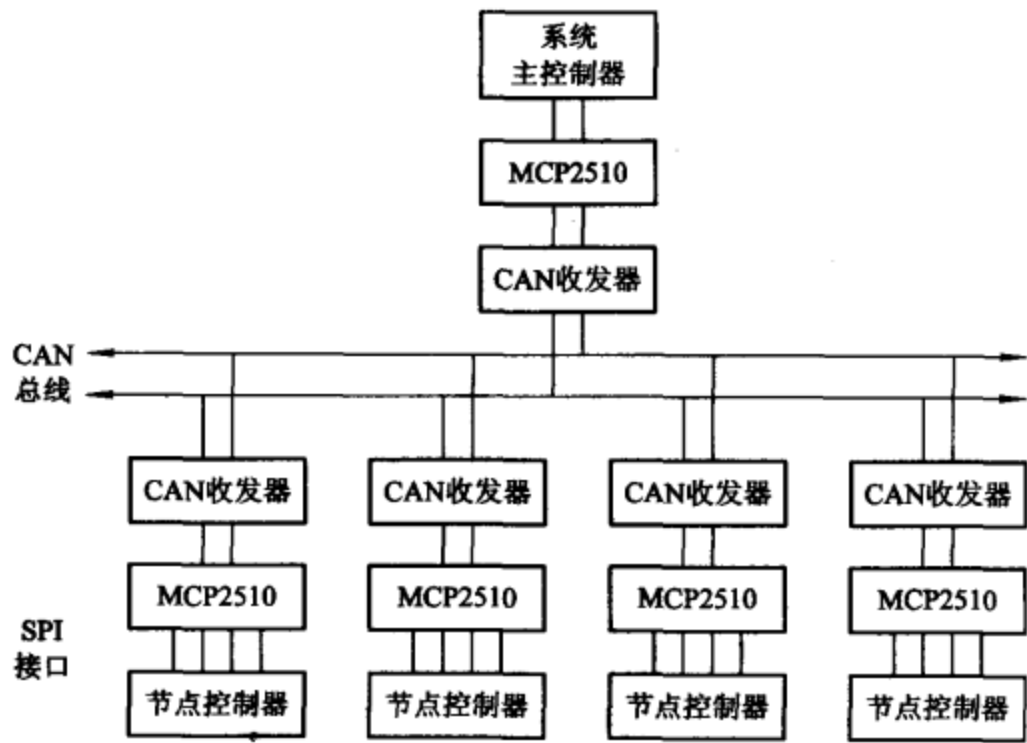


图 6-10 试验用的 CAN 网络结构图

在图 6-10 网络结构中，系统主控机可有一台或多台，相当于上位机，负责系统的总体管理，可以向网络节点发送命令，接受节点数据，进行存储、分析、统计、显示与打印等工作。在系统主控机内装有 PC CAN 网卡(CAN 总线通信接口适配卡)。系统主控机中具有以下几种软件模块：

- (1) 初始化模块。该模块用来设置 CAN 网络通信速率、输出控制方式、报文标识符屏蔽格式等参数，设定各节点工作空间的位置与规模，初始化节点缓冲区域的缓冲文件等。
- (2) 采集节点信息与数据模块。该模块可以采用指定节点发送、由节点主动发送和自动轮询采集等多种方式，并可以统计与分析数据采集的进度与状态。
- (3) 向节点发送参数、命令或程序模块。
- (4) 数据统计分析模块。
- (5) 系统管理模块。

图 6-10 所示的网络结构中有 12 个网络节点，每一节点都通过传感器采集现场的有关数据。在每一节点电路中都配置了 MCP2510 CAN 控制器和 TJA1050 总线收发器，其接口电路如图 6-11 所示。系统主控机与网络节点采用双绞线连接，实验时最大节点间的距离为 35 m。

【问题 1】(3 分)

CAN 总线能够使用光纤和(1)等多种传输媒体。总线信号以差分电压传送，两条信号线分别为 CAN_H 和 CAN_L。静态时这两条信号线均为 2.5 V，此状态表示(2)，也可以称之为“隐性”。CAN_H 比 CAN_L 高，表示逻辑“0”(或称为“显性”)，此时，通常电压值为 CAN_H=3.5 V，CAN_L=1.5 V。当“显性”位和“隐性”位同时发送时，最后总线数据将为(3)。

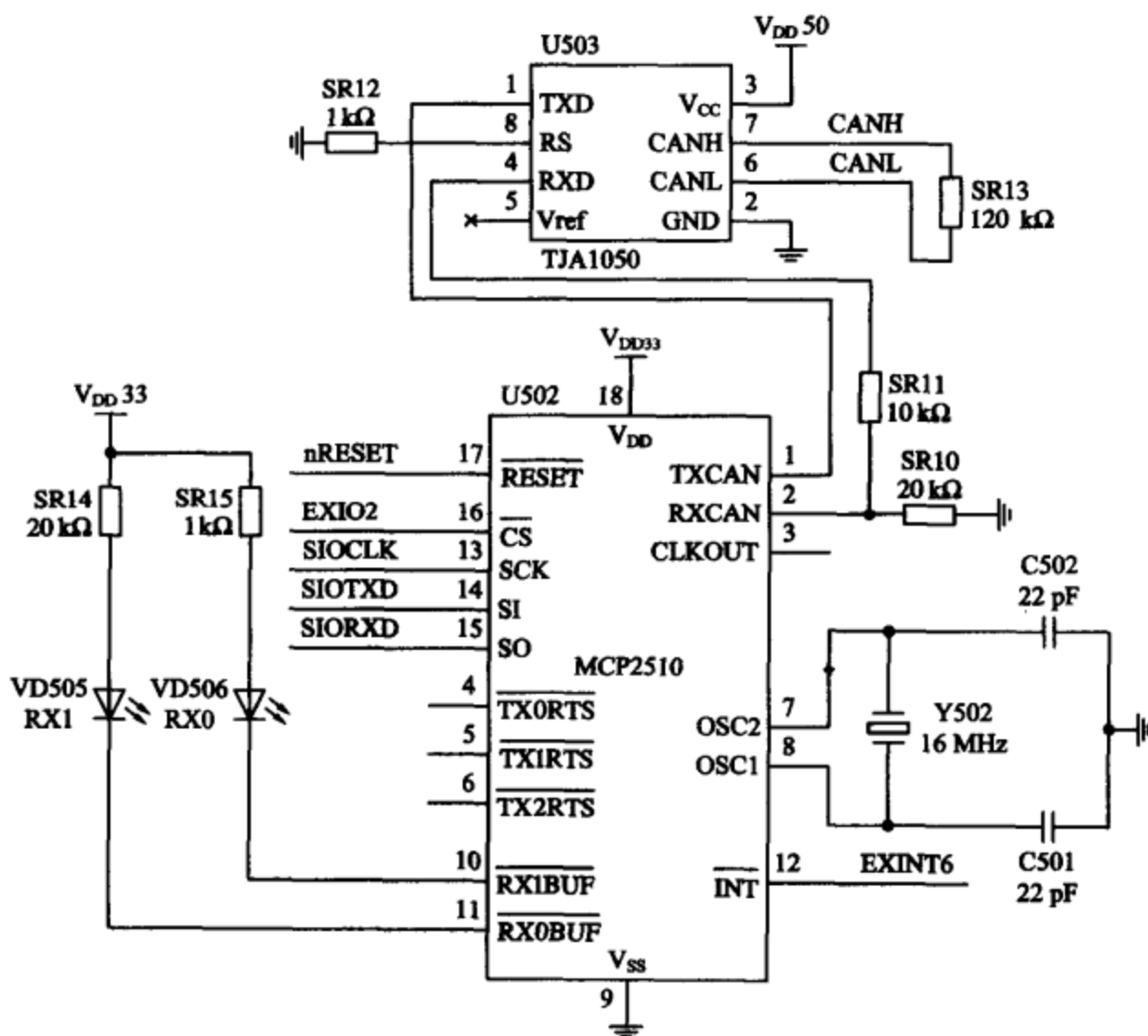


图 6-11 MCP2510 组成的 CAN 接口电路

【问题 2】(3 分)

CAN 现场总线具有如下基本特征：

① CAN 总线是一种多主方式的串行通信总线，当 CAN 信号传输距离在 40 m 以内时，其最高通信速率为 1 Mb/s；当信号传输距离在 10 km 以内时，CAN 总线仍可提供 50 kb/s 的数据传输速率。网络上节点的数目主要取决于总线驱动电路，目前可达 110 个节点。

② 通信的灵活性。CAN 总线允许采用多主方式工作，网上任一节点均可在任何时刻主动向网上其他节点发送信息(不分主从)，无需站地址等节点信息。事实上，通信取决于报文标识符进行(在 CAN2.0A 标准中规定了 2032 种报文标识符)，采用报文滤波即可实现点到点、一点到多点或者全广播等多种方式通信，无需专门调度。

③ 通信的实时性。CAN 网络上节点的信息可分为不同的优先级，从而能满足不同的实时性要求。

④ 通信的可靠性。CAN 采用了非破坏性的总线仲裁技术，当多个节点同时向总线发送信息时，优先级较低的节点会主动退出发送，具有相对最高优先级的节点可以不受影响地继续传输数据，从而大大节省了总线仲裁时间，特别是在网络负荷很重的情况下也不会引起网络瘫痪。另一方面，CAN 协议规定了采用短帧结构(比如一个数据帧内有效数据为

8 个字节), 帧传输时间短, 抗电磁干扰能力强。每帧内有 CRC 校验及其他校验措施(适用于位数小于 127 位的帧)。当节点发现严重错误时, 能自动关闭输出, 使其他节点操作不受影响。

基于以上基本特征, 请估算具有最高优先权的数据帧最快可在多长时间内获得传输权限。请在 150 字以内简要说明理由并列计算出计算过程。

【问题 3】(4 分)

在图 6-11 所示的电路图中, MCP2510 芯片组成的 CAN 控制器和 TJA1050 芯片所组成的电路各有何作用。

【问题 4】(2 分)

MCP2510 和 TJA1050 连接的两个信号都是单向信号。在图 6-11 所示的电路图中, MCP2510 芯片输入信号即 RXCAN 高电平的范围 V_{ih} 是 2~4.3 V。这并不能满足 5 V 逻辑的 TJA1050 芯片的输出电平, 因此需要进行 I/O 接口电路的电平转换。在图 6-11 所示的电路图中, 电阻 SR10、SR11 的阻值在选择时需要考虑哪些因素?

【问题 5】(3 分)

在图 6-10 所示的网络结构中, 每一个网络节点相对于 CAN 控制器而言, 有一个隐含的 CAN 总线接口通信模块, 上电后能自动完成接口的初始化工作, 并进入通信监测工作状态。请在 150 字以内简要列出每一个网络节点中至少还应当包括哪些功能模块。

6.1.3 要点解析

1. 上午试题要点解析

(1) B。用流水线时钟周期至少为 8 ns; 若不采用流水线则需要的时间为 $(6+8+5+7+6)=32$ ns。此流水线的最大加速比 S_p 为 $32/8=4$ ($0 \leq S_p \leq p$, 其中 p 表示流水线级数)。

(2) B。PPU I/O 控制方式一般用于高效的大型系统中, PPU 是指外围处理机方式。

(3) C。利用公式 $MTBF = \frac{1}{\lambda}$ 可分别计算出 A、B 这两个子系统的 λ_a 、 λ_b 和整个系统的 λ 。

$$\lambda_a = \frac{1}{200h} = 0.005, \lambda_b = \frac{1}{400h} = 0.0025, \lambda = \frac{1}{100h} = 0.01$$

然后再根据公式 $\lambda = \lambda_a + \lambda_b + \lambda_c$ 计算出

$$\lambda_c = \lambda - \lambda_a - \lambda_b = 0.01 - 0.005 - 0.0025 = 0.0025$$

最后利用公式 $MTBF = \frac{1}{\lambda}$ 计算出子系统 C 的 MTBF 值 = 400 h。

(4) C。系统可靠度计算, 并联系统: $1 - (1 - R_1)(1 - R_2) = R_1 + R_2 - R_1 R_2$, 串联系统: $R_1 R_2$, 其中 R_1 、 R_2 分别指单个系统的可靠度。本题 $R = 0.95 \times (0.95 + 0.95 - 0.95 \times 0.95) \times (0.95 + (0.95 + 0.95 - 0.95 \times 0.95) - 0.95 \times (0.95 + 0.95 - 0.95 \times 0.95)) = 0.95 \times 0.9975 \times 0.9999 = 0.9984^2 = 0.9475$

(5) B。Gantt 图不能表现各个活动的顺序和它们之间的因果关系, 但可以表现哪些活动具有并行性。Gantt 图可以表现各个活动的起始时间, 也可以表现各个活动完成的进度。

(6) C。“工程计划图”又称为工程网络图, 与 Gantt 图(因果分析图, 又称鱼骨刺图)是两种常用的项目管理工具, 用于工作计划安排的评审和检查。本试题中该工程计划图的关键路径是 $A \rightarrow B \rightarrow D \rightarrow F \rightarrow G$, 该路径所花费的时间最长(20 天)。由于作业 F 是关键路径中

的一个作业,因此作业 F 的松弛时间为 0。由于松弛时间=最迟开始时间-最早开始时间,而该工程计划图中作业 F 的完成依赖于作业 C、D 的完成,虽然作业 B、C 可并行进行,但作业 A、B、D 需花费 12 天时间,作业 F 最早可在第 13 天开始。因此作业 F 最迟开始时间是第 13 天。

(7) B。ISO9000 资质认证过程中要对企业的各方面进行严格审查,还要每年进行自检和外检。ISO9000 质量管理体系认证书的有效期为三年。

(8) C。同事张三、小李为本企业合作开发一套库存管理信息系统,该系统已通过验收并投入使用。3 年后,小李辞职,在 Internet 网上申请了一个个人网站,为丰富网站内容并宣扬个人工作业绩,小李将该管理软件上传至个人网站的网友下载区中。小李的这种行为至少侵犯了信息网络传播权。

(9) D。X.509 数字证书格式中包含的元素有证书版本、证书序列号、签名算法标识、证书有效期、证书发行商名字、证书主体名、主体公钥信息和数字签名。可见数字证书主体公钥被放置于证书中且对外公开。

(10) C。防火墙技术是建立在内外网边界上的过滤封锁机制,能够防止外网未经授权访问内网,能够防止外网对内网的攻击,也能防止内网未经授权访问外网。但是,仅使用防火墙不能有效地防止内网的攻击,通常不具备反向追踪网络攻击的能力。

(11) C。通常侵入被攻击主机的木马程序属于服务程序。攻击者一旦获取了木马程序的 IP 地址和端口号,便与它进行连接。为了检测系统是否有木马侵入,可以使用 Windows 操作系统的 netstat -an 命令查看当前的活动连接端口。通常可利用防火墙的包过滤功能有效地防止外部非法连接来拦截木马。

其他选项中,命令 ipconfig /all 用于查看计算机的网络配置信息。环回测试命令 ping 127.0.0.1 用于初步检测计算机网卡是否处于正常工作状态。命令 nslookup 命令用于查询域名与 IP 地址的对应关系。

(12) C。由于页面的大小为 $4\text{ KB}=2^2 \times 2^{10}=4096\text{ B}$,逻辑地址 $8644=4096 \times 2+452$,得页号为 2,偏移量为 452。查表 7-15 可知,页号为 2 所对应的物理块号是 8,因此所求物理地址 $=4096 \times 8+452=33\,220$ 。

(13) A。当像素颜色深度为 24 位,且不采用压缩算法时,存储一个像素需要 24 位,即占用 3 个字节存储空间。一张照片的像素数为 1600×1200 ,则存储该幅图像所需的存储空间为 $1600 \times 1200 \times 3\text{ B}=5\,760\,000\text{ B}$ 。最多可以存储的照片数目为

$$N = \frac{256 \times 1024 \times 1024\text{ B}}{1600 \times 1200 \times 3\text{ B}} = 46.604$$

取整数得 $N=46$ 。

(14) D。视频卡中的信号获取器将输入的彩色全电视信号经 A/D 变换和数字解码后在窗口控制器的控制下实时地存入帧存储器中。其中帧存储器的信号为数字的 YUV 信号。

(15) D。CMM 是对软件组织进化阶段的描述,随着软件组织定义、实施、测量、控制和改进其软件过程,软件组织的能力经过这些阶段逐步前进。CMM 将软件过程的成熟度分为五个等级,详见表 6-7。

表 6-7 CMM 模型的五个等级

级别	描述	特 点	关键过程域
第一级	初始级	软件过程是无序的,几乎没有明确定义的步骤,成功完全依赖个人努力和英雄式的核心任务;企业一般不具备稳定的软件开发与维护的环境,常在遇到问题的时候就放弃原定的计划,而只专注于编程与测试	
第二级	可重复级	在这一级别上,建立了基本的项目管理过程来跟踪成本、进度和机能,制定了必要的过程纪律,并基于以往的项目经验来计划与管理新的项目	需求管理、软件配置管理、软件子合同管理等
第三级	定义级	管理和工程的软件过程已经文档化、标准化,并综合成整个软件开发组织的标准软件过程。所有的项目都采用根据实际情况修改后得到的标准软件过程来发展和维护软件	组织过程定义、集成软件管理、软件产品工程等
第四级	定量管理级	在这一级别上,制定了软件工程和产品质量的详细度量标准,使用定量分析来不断地改进和管理软件过程。软件过程和产品的质量都被开发组织的成员所理解和控制,因此软件产品具有可预期的高质量	定量的过程管理、软件质量管理
第五级	优化级	通过来自过程质量和新观念、新技术的反馈使过程能持续不断地改进。可见整个企业将会把重点放在对过程进行不断的优化上。企业会采取主动去找出过程的弱点与长处,以达到预防缺陷的目标	缺陷预防、技术变更管理、过程变更管理

(16) B。软件开发包括需求分析、设计、编码、测试和维护等阶段。瀑布模型将软件生命周期划分为制定计划、需求分析、软件设计、程序编写、软件测试和运行维护六个基本活动,并且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。瀑布模型强调文档的作用,并要求每个阶段都要仔细验证。结构化开发方法的生存周期划分与瀑布模型相对应,因此也是与其最相适应的软件开发方法。

对于本题的选项 A,OMG 描述构件的定义为,构件是一个物理的、可替换的系统组成部分,它包装了实现体且提供了对一组接口的实现方法。构件化方法是以过程建模为先导、以构架为中心、基于构件组装的应用系统开发方法。它可以裁剪为面向对象方法,适合于构件组装模型。构件化方法和面向对象方法鼓励构件组装(复用),面向对象过程沿演化的螺旋迭代,因此它们与瀑布模型软件开发思路不相适应。

对于选项 C,对象是指由数据及其容许的操作所组成的封装体。所谓面向对象,就是基于对象概念,以对象为中心,以类和继承为构造机制来认识、理解、刻画客观世界和设计、构建相应的软件系统。面向对象方法是一种把面向对象的思想应用于软件开发过程中,指导开发活动的系统方法,简称 OO (Object Oriented)方法。

对于选项 D,快速原型模型的第一步是建造一个快速原型,实现客户或未来用户与系

统的交互，用户或客户对原型进行评价，进一步细化待开发软件的需求。通过逐步调整原型使其满足客户的要求，开发人员可以确定客户的真正需求是什么；第二步则在第一步的基础上开发客户满意的软件产品。显然，快速原型方法可以克服瀑布模型的缺点，减少由于软件需求不明确而带来的开发风险，具有显著的效果。

(17) A。根据国家标准 GB 8566—88《计算机软件开发规范》的规定，单元测试是根据详细设计阶段给出的规格说明书在编码阶段完成的测试工作；集成测试的计划是在概要设计阶段制定的；系统测试计划应该在需求分析阶段就开始制定，并在设计阶段细化和完善，而不是等系统编码完成后才制定测试计划；而验收测试则用来检测产品是否符合最终用户的需求。

软件测试的各个阶段与软件开发阶段的对应关系如图 6-12 所示。

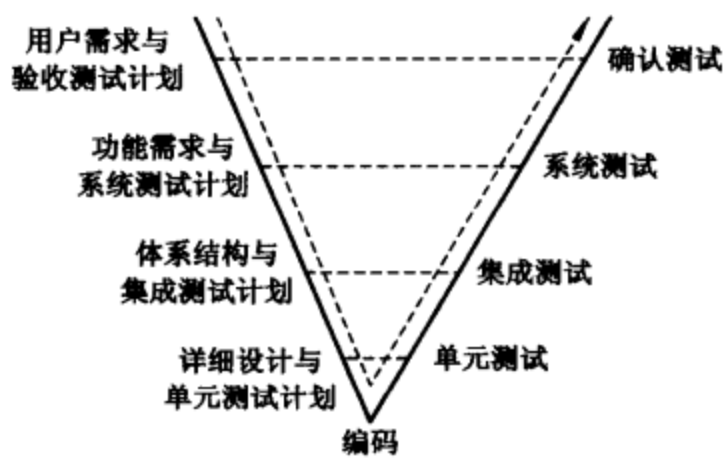


图 6-12 各软件开发阶段与软件测试阶段的对应关系图

(18) B；(19) C。继承机制是面向对象技术提供的一种解决软件复用问题的途径。其访问控制包括 public(公有派生)、protected(保护派生)和 private(私有派生)。基类中的访问权限在基类和派生类中的继承关系如表 6-8 所示。

表 6-8 访问权限的继承关系

派生性质	在基类中的访问权限	在派生类中的访问权限
public (公有派生)	public	public
	protected	protected
	private	不可访问
protected (保护派生)	public	protected
	protected	protected
	private	不可访问
private (私有派生)	public	private
	protected	private
	private	不可访问

对于(18)空缺处，① 由于类 R 中定义了一个私有方法 F1。类的私有成员只有在本类中才能访问，因此选项中凡出现“可以访问 F1”都是错误的，据此排除选项 A、C。

② S 是采用 public 继承方式从 R 派生而来的，那么类 R 中的所有公有成员都是 S 的

公有成员，在程序中的任何位置均可被访问，因此选项 B 是正确答案。

对于(19)空缺处，① 通过继承，F1、F2、F3 都成为类 T 的成员。

② 由(18)空缺处的分析可知，F1 不可以被外界访问，因此选项中凡出现“可以访问 F1”都是错误的，据此排除选项 A。

③ 由于 T 采用的是 private 继承方式，S 中的成员都成为 T 的 private 成员，即 F2、F3 都是 T 的 private 成员。

④ 私有成员只有本类可以访问，所以 T 的对象不能访问 F2 和 F3，只有 T 的成员才能访问它们。因此选项 C 是正确答案。

(20) D。四维立方体单级互联函数(Cube3)的运算规则是：将已知的微处理器编号从右边第 0 位起数到第 3 位，将该位求反即可得到所求的微处理器编号。因此对于第 5 号微处理器，其二进制编码为 0101，经过变换后为 1101，即与第 13 号微处理器相连接。

注意：如果本试题所采用的是全混洗单级互联函数 Shuffle(无进位左循环一位的函数)，那么第 5 号微处理器的编码(0101)经变换后为 1010，即为第 10 号微处理器。

(21) A。逻辑函数 $F=A \oplus B$ 和 $G=A \odot B$ 的真值表如表 6-9 所示。

表 6-9 逻辑函数 $F=A \oplus B$ 和 $G=A \odot B$ 的真值表

A	B	F	G
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

对比表 6-9 中最后两列的逻辑变量值可得出，异或逻辑函数 $F=A \oplus B$ 和同或逻辑函数 $G=A \odot B$ 满足关系 $F=\bar{G}$ 。

(22) B。选项 A 的最小项之和是：

$$Y_1(A, B, C) = A + \bar{B}C = A\bar{B}\bar{C} + A\bar{B}C + ABC + \bar{A}BC;$$

选项 C 的最小项之和是：

$$Y_3(A, B, C) = ABC + \bar{A}\bar{B}\bar{C} = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + ABC + \bar{A}BC$$

选项 D 的最小项之和形式是：

$$\begin{aligned} Y_4(A, B, C, D) &= ACD + \bar{A}\bar{C}D + AB\bar{D} \\ &= \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + A\bar{B}\bar{C}D + AB\bar{C}\bar{D} + ABC\bar{D} + ABCD \end{aligned}$$

(23) B。集电极开路输出的 TTL 与非门也称为 OC 门，其输出端可直接与其他 OC 门输出端相连，实现“线与”。

(24) C。由 n 位触发器构成的扭环形计数器其有效状态数有 $2n$ 个，无关状态数有 $2^n - 2n$ 个。

(25) B。RISC 是精简指令集计算机的英文缩写，它是在继承复杂指令集计算机(CISC)成功技术并克服其缺点的基础上产生并发展起来的，RISC 与 CISC 在特点上的区别如表 6-10 所示。

表 6-10 RISC 与 CISC 在特点上的区别

类别	RISC	CISC
指令系统	指令长度固定, 指令种类少(少于 100 种)	指令数量很多
执行时间	选取使用频率较高的一些简单指令, 且指令执行时间较短	有些指令执行时间很长, 例如整块的存储器内容复制
编码长度	固定, 通常为 4 个字节	可变, 1~15 字节
寻址方式	种类少, 即简单寻址	种类丰富
操作	设置大量通用寄存器, 访问存储器指令简单, 只能对寄存器进行算术和逻辑操作	可以对存储器和寄存器进行算术和逻辑操作
编译	采用优化编译技术, 生成高效的目标代码程序	难以用优化编译器生成高效的目标代码程序

由上表可知, RISC 计算机设置大量通用寄存器, 访问存储器指令简单, 只能对寄存器进行算术和逻辑操作。

(26) B. 片上系统(SOC)是指在单芯片上集成数字信号处理器、微控制器、存储器、数据转换器及接口电路等电路模块, 可以直接实现信号采集、转换、存储和处理等功能, 其中, 知识产权核(IP Core)设计是 SOC 设计的基础。

IP 核模块有行为、结构和物理三级不同程度的设计, 对应描述功能行为的不同可分为 IP 软核、IP 硬核和 IP 固核三类, 它们各自的定义及其优缺点见表 6-11。

表 6-11 IP 核 模 块

类别	描 述	优 缺 点
IP 软核	通常是用硬件描述语言文本形式提交给用户, 它经过 RTL 级设计优化和功能验证, 但其中不含有任何具体的物理信息	用户可以综合出正确的门电路级设计网表, 并可以进行后续的结构设计, 具有很大的灵活性; 其主要缺点是缺乏对时序、面积和功耗的预见性, 且 IP 软核是以源代码的形式提供的, IP 知识产权不易保护
IP 硬核	基于半导体工艺的物理设计, 已有固定的拓扑布局 and 具体工艺, 并已经过工艺验证, 具有可保证的性能	可以以电路物理结构掩膜版图和全套工艺文件的形式提供给用户, 无需提供寄存器转移级(RTL)文件, 更易于实现 IP 保护; 其缺点是灵活性和可移植性差
IP 固核	设计程度介于软核和硬核之间, 除了完成软核所有的设计外, 还完成了门级电路综合和时序仿真等设计环节	一般以门级电路网表的形式提供给用户

由以上分析可知, 本试题中关于知识产权核的说法错误的是选项 B, “基于半导体工艺的物理设计, 已有固定的拓扑布局 and 具体工艺, 并已经过工艺验证, 具有可保证的性能”是对 IP 硬核的描述。

(31) C。通常嵌入式文件系统比较简单, 主要提供文件存储、检索和更新等功能, 一般不提供保护和加密等安全机制。它以系统调用和命令方式提供以下 3 种文件的操作服务:

- ① 提供建立、修改、改变和删除目录等服务;
- ② 提供创建、打开、读写、关闭和撤销文件等服务;
- ③ 设置、修改对文件和目录的存取权限。

通常, 嵌入式文件系统可支持 FAT32 等标准的文件系统, 还可支持自定义的实时文件系统和多种存储设备, 具有可裁减和可配置等特点。

(32) B。通常, Ethernet 以太网的数据帧最大长度为 1518 字节, Ethernet II 帧头为 18 字节, IP 头最少为 20 字节, TCP 头最少为 20 字节, 数据部分最长为 $1518 - 18 - 20 - 20 = 1460$ B。

(33) B。在 TCP/IP 网络中, 传输层的传输控制协议 TCP 和用户数据包协议 UDP 的源端口、目的端口均占用 16 bit, 其所能表达的端口号范围达 $2^{16} = 65\,536$, 但其中为各种公共服务保留的 TCP/UDP 端口号范围是 1~1023。

(34) B。IEEE 802.11 标准主要使用红外线、跳频扩频与直接序列扩频三种传输技术。蓝牙技术采用跳频扩频技术和时分多路利用技术可使特定移动电话、便携式电脑、便携式通信设备的主机之间在近距离内(通常为 10 m 以内)实现无缝的资源共享。

(35) B。由于实时系统的应用环境具有特殊性, 因此其存储管理与普通操作系统的存储管理有很大的区别。其中, 最主要的区别是实时操作系统不采用虚拟内存管理方式, 而是采用物理内存直接管理方式。理由如下:

① 虚拟存储技术用于提供比计算机系统中实际使用的物理内存大得多的内存空间, 使编程人员在写程序时不用考虑计算机中的物理内存的实际容量。其机制中换页操作及其查询、调度等增加了系统的复杂性, 减慢了响应时间。

② 实时系统软件小巧紧凑, 因此运行过程中需要调入内存的数据比较少。

③ 实时系统一般都运行在单用户的模式下, 无需在软件中强调系统的安全机制。

(36) A。大端字节顺序存储法是将高字节数据存放在低地址处, 低字节数据存放在高地址处, 因此选项 A 是答案。而小端字节顺序存储法是将低字节数据存放在低地址处, 高字节数据存放在高地址处, 例如选项 C。

(37) B; (38) A。本题考查虚拟存储管理系统的知识。虚拟存储管理系统的基础是程序的局部性原理。工作集是指在某段时间间隔内, 进程所要访问的页面集合。虽然程序只需少量的几页内存就可以运行, 但为了使程序更有效地运行, 必须使程序的工作集全部在内存(主存储器)当中, 否则会导致进程在运行中频繁出现缺页中断, 从而出现频繁的页面调入/调出现象。

在 t_1 时刻的任务工作集的计算方法是: 从 t_1 时刻开始往回数 10 个(窗口尺寸)页面, 即 {4 7 5 1 7 6 2 7 3 4}, 然后把这个窗口中重复的页面去除, 就可得到相应的工作集为 {1, 2, 3, 4, 5, 6, 7}。

(39) B。通过文件 1 的目录项可知, 第 1 个逻辑块存放在第 2 个物理块中, 然后查询 FAT 表可知, 第 2、3、4 个逻辑块分别存放在第 5、6、7 个物理块中。第 7 个物理块中 0xFFFF 是个特殊的值, 表明文件的结束, 所以该文件总共有 4 个物理块。

(40) B; (41) A。冒泡排序的过程很简单。首先将第1个数与第2个数相比较,若为逆序则交换两数,然后比较前两个数与第3个数,依次类推,直到将第 $n-1$ 个数与第 n 个数比较完为止。上述过程称为一趟冒泡排序,结果是最大的数被排到了最后。然后进行第2趟,对前面 $n-1$ 个数进行冒泡排序,结果是次大的数被移到了 $n-1$ 的位置上。一般来说,第 i 趟冒泡排序是将第1个数排到了第 $n-i+1$ 的位置上,整个排序过程需进行 $k(1 \leq k \leq n)$ 趟。

分析冒泡排序的效率时,若初始序列为正序,则只进行一趟排序,在排序过程中只进行 $n-1$ 次比较,不交换数据;若为逆序,则需进行 $n-1$ 趟排序,需进行 $n(n-1)/2$ 次比较,交换数据的数量组也相同。因此,冒泡排序的复杂性是 $O(n^2)$ 。快速排序是对冒泡排序的一种改进,其基本思想是通过一趟排序将待排序的数据分成两部分,其中一部分的关键字均比另一部分的关键字小,然后再对这两部分分别进行快速排序,最后达到整个序列有序的目的。快速排序的复杂性是 $O(n \lg n)$ 。

将题中给定的整数序列(541, 132, 984, 746, 518, 181, 946, 314, 205, 827)进行从小到大排序,若先选出较大的元素,则对于冒泡排序,第1趟操作为 $541 \leftrightarrow 132$, $984 \leftrightarrow 746$, $984 \leftrightarrow 518$, $984 \leftrightarrow 181$, $984 \leftrightarrow 946$, $984 \leftrightarrow 314$, $984 \leftrightarrow 205$, $984 \leftrightarrow 827$,其结果得到的序列为(132, 541, 746, 518, 181, 946, 314, 205, 827, 984);对于直接选择排序,第1趟操作为 $984 \leftrightarrow 827$,其结果得到的序列为(541, 132, 827, 746, 518, 181, 946, 314, 205, 984)。

采用快速排序(以中间元素518为基准)的第1趟扫描结果是(205, 132, 314, 181, 518, 746, 946, 984, 827)。

(42) A。减小干扰源的 du/dt 主要是通过干扰源两端并联电容来实现的。减小干扰源的 di/dt 则是在干扰源回路串联电感或电阻及增加续流二极管来实现的。

(43) A。这是一段子程序调用(BL指令完成)的程序,子程序DO ADD完成加法运算,操作数放在 $r0$ 和 $r1$ 寄存器中,结果放在 $r0$ 中。

(44) B。 $R13$ 也被称为SP指针,被用做栈指针,通常在系统初始化时需要对所有模式的SP指针赋值。

(45) A。ARM是典型的RISC体系,其模式控制位 $M[4:0]$ 为0b10010时,代表IRQ模式; $M[4:0]$ 为0b10111时,代表Abort模式; $M[4:0]$ 为0b10011时,代表Supervisor模式; $M[4:0]$ 为0b11111时,代表System模式。

(46) C。当禁止存储管理单元(MMU)时,先要确定芯片是否支持cache和write buffer。如果芯片规定禁止存储管理单元(MMU)时,禁止cache和write buffer,则存储访问将不考虑C和B控制位。

如果芯片禁止存储管理单元(MMU)时,可以使能cache和write buffer,则数据访问时 $C=0$ 和 $B=0$;读取指令时,对于分开的TLB, $C=1$,对于统一的TLB, $C=0$ 。

(47) D。嵌入式系统支持的内存块由大到小分别是段(1 MB)、大页(64 KB)、小页(4 KB)和极小页(1 KB)。大页可分成大小为16 KB的子页,小页可分成大小为1 KB的子页。

(48) C。在ARM体系结构的嵌入式系统中,快速上下文切换技术控制位编码为0b10,则表示方位类型是保留;控制位编码为0b00,则表示方位类型是没有访问权限;控制位编码为0b01,则表示方位类型是客户类型;控制位编码为0b11,则表示方位类型是管

理者权限。

(49) A。NAND Flash 的读取速度比 NOR Flash 稍慢一些,其随机读取能力差,适合大量数据的连续读取。

(50) B。SRAM 的写操作周期是:① $CE=0$,启用该芯片;② $R/W'=0$;③ 地址出现在 address 线上,数据出现在 data 线上。

(51) C。嵌入式系统由硬件部分和软件部分构成。通常将嵌入式软件分为嵌入式系统软件、嵌入式应用软件和嵌入式支撑软件三大类,如表 6-14 所示。交叉编译器是嵌入式系统的辅助工具软件。

表 6-14 嵌入式软件分类

软件类型	功 能	例 子
系统软件	控制和管理嵌入式系统资源,为嵌入式应用提供支持的各种软件	嵌入式操作系统、设备驱动程序及嵌入式中间件等
应用软件	定义嵌入式设备的主要功能和用途,并负责与用户进行交互	MP3 播放软件、手机软件、路由策略软件及恒温控制软件等
支撑软件	辅助软件开发的工具软件	在线仿真工具、FPGA 编程软件、交叉编译器及源程序模拟器等

(52) D。在 C 语言环境中,结构体类型说明的一般形式:

`struct 结构体名{类型名 结构体成员名...}结构体变量名;`

其中,“struct”是结构体类型的关键字。对于本题而言,example 是结构体变量名,而非结构体类型名。

(53) A。fork/exec 模型和 spawn 模型创建任务时,都要先为新任务分配相应的数据结构,然后再为新任务分配内存空间。

这两种模型的主要差别在于内存的分配方式,在 fork/exec 模型中,首先调用 fork 函数为新任务创建一份与父任务完全相同的内存空间,然后再调用 exec 函数装入新任务的代码,并覆盖原父任务的内容。而 spawn 模型则直接为子任务分配一个全新的地址空间,然后再将其代码装入运行。

(54) B。在实时操作系统中,任务间通信是指任务之间为了协调工作,需要相互交换数据和控制信息,可分为低级通信和高级通信两种类型。PV 操作等信号量机制和信号机制属于低级任务间通信方式。消息传递、共享内存、管道通信属于高级任务间通信方式。

任务间通信又可分为直接通信和间接通信两种类型。send、receive 原语属于直接通信方式;邮箱、消息队列属于间接通信方式。

(55) A。实模式的主要特点是:① 不再划分“系统空间”和“用户空间”,整个系统中有一个物理内存地址空间;② OS 内核与外围应用程序之间不再有物理边界;③ 系统中“任务”或“进程”全都是内核线程;④ 只有运行上下文和栈是独享内核线程,其他资源都是共享的。

(56) A. 将逻辑地址 0xE9B7 转换成二进制形式为(1110 1001 1011 0111)₂。

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	1	0	0	1	1	0	1	1	0	1	1	1
逻辑页面号								页内偏移量							

页面大小为 1 KB, 对于逻辑地址 0xE9B7 所对应的逻辑页面号为 0x3A, 页内偏移量为 0x1B7。

(57) A. Linux 系统的五种基本文件类型是普通文件、管道文件、设备文件、链接文件和目录文件。其中, 管道文件又称为 FIFO 文件, 主要用于进程间传递数据。

(58) D. Linux 系统文件访问的三种处理模式是读取 r、写入 w 和执行 x。对普通文件默认的访问权限是: 文件所有者能够读写, 但不能执行; 文件所有者同组成员和其他成员只能读取, 即“-rw-r--r--”。

(59) B. 在 Linux 系统 smb.conf 配置文件中, 共享的安全级别有四种: 共享 share 安全级、用户 user 安全级、服务器 server 安全级及域 domain 安全级。其中, 文件和打印机共享默认的安全级别是用户 user 安全级。

(60) D. 嵌入式系统的设计和开发流程一般分为以下几个阶段: 产品定义(即系统需求分析阶段、规格说明阶段)、硬件和软件划分、迭代与实现、详细的硬件与软件设计、硬件与软件集成、系统测试和系统维护与升级。

(61) D. 模块的耦合度表现了模块之间相互关联的程度, 从低到高分分为无直接耦合、数据耦合、标记耦合、控制耦合、公共耦合和内容耦合等六级。

(62) B. 一个故障已经被发现, 而且也被排除了, 因为在修改过程中纠正了旧的错误又会引入新的错误, 此时应该进行退化测试(Regression Test), 以防止出现新的错误。退化测试包括以下步骤: ① 插入新代码, 程序成为新版本; ② 测试可能受新代码影响的功能; ③ 测试修改前的基本功能; ④ 测试新版本的功能。

对于选项 A, “程序走查”, 是由 Bell 等人提出的一种评价程序语言的方法, 它以用户为中心, 通过走查用户使用程序完成任务的情况进而评价程序语言设计的可用性和流畅性。

对于选项 C, “软件评审”, 是指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。

对于选项 D, “软件接收测试”, 是保证关键任务的应用软件能在其所运行的硬件设施上高效地运行, 与严格的操作标准一致。

(63) C. 嵌入式应用软件的开发流程与通用软件的开发流程大体相同, 但在开发所使用的设计方法上有一定的差异。整个软件的开发流程可分为在软硬件划分阶段确定硬件驱动接口阶段、软件功能模块按照实时性进行划分阶段、各软件功能模块的代码生成阶段、软件功能模块的集成测试阶段、代码固化及固化后的调试阶段。

(64) C. 嵌入式系统需求分析的任务通常包括确定功能要求、分析数据要求、修正开发计划等, 而软硬件模块划分工作属于硬件与软件划分阶段的工作内容。

(65) A. 系统维护工作流程中正确的顺序是: 用户提交维护申请报告→核实和评价维

护申请报告→制定维护计划→实施维护→测试→更新文档→交付使用。

(66)~(70) B、C、A、B、D。

参考译文：每天发生成千上万次的电子交易，这使得商业机构产生了保护电子交易数据完整性和确认交易数据标识的迫切需求。从而导致整个行业都在探索更好更安全的控制 IT 操作的方法，探索在整个网络基础结构和客户设备中深入而广泛地部署增强安全机制的方法。一种能够实现安全社区的有效设想是在客户机硬件中开发基于标准的安全芯片。在硅片中封装核心安全能力、在机器操作的最深层次集成安全设施的理论，对用户和对那些负责 IT 安全运营的人员都有极大的好处。

(71)~(75) B、C、A、D、A。

参考译文：无诱饵钓鱼欺骗是一种诡秘的做法，其中恶意代码会装入个人计算机或服务器，在不知不觉中或者未经用户同意就将用户错误地引向骗人的网站。它被叫做“无诱饵钓鱼”。

在钓鱼式欺骗中，为了从受信个人那里获得个人和财务信息，作恶者发送似乎是合法的电子邮件，看上去它来自某个最流行的网站。而在无诱饵钓鱼欺骗中，由于它不是一个针对个人，受害方丝毫不会怀疑这些行为，因此会有大量的计算机用户受害。其中一种无诱饵钓鱼欺骗的形式就是，随电子邮件发出的(恶意)代码修改个人计算机上的本机主文件。该主文件将 URL 转换成计算机访问网站的数码串。拥有修改过的主文件的计算机去假冒网站，即使用户键入了正确的因特网地址或者点击了受影响的书签条目。有些消除间谍件的程序能纠正此种破坏，但除非用户改变了浏览习惯，否则它还是会频繁发生。

2. 下午试题要点解析

试题一

【问题 1】(2 分)

这是一道要求读者通过阅读电路原理图说明相关信息的基本常识题。本题的分析思路如下：

① 图 6-7 所示的单极性输入电路原理图中，“ANALOG INPUTS”就是模拟电压输入的意思；

② 从图 6-7 所示的电路原理图中可知，AD574A 有两种额定的模拟电压输入范围，0~10 V 的输入接在第 13 脚和第 9 脚；0~20 V 的输入接在第 14 脚和第 9 脚。

③ 如果是双极性模拟电压输入，则第 13 脚用于输入±5 V 范围内的电压信号；第 14 脚用于输入±10 V 范围内的电压信号。

【问题 2】(4 分)

这是一道要求读者掌握 A/D 转换电路具体元器件作用的分析题。本题的解答思路如下：

① 在图 6-7 所示的单极性输入电路原理图中，电位器 R1 的调整端通过一个 100 kΩ 的电阻连接至 AD574A 芯片的第 12 脚(BIP OFF)。

② 查阅图 6-6 可知，AD574A 芯片的第 12 脚功能的英文注释是“BIPOLAR OFFSET”，即偏移量调整。由此可知，电位器 R1 用于 A/D 转换电路的偏移量调整。为了

使 A/D 转换电路的量化误差为 $\pm 1/2\text{LBS}$, AD574A 芯片的额定偏移规定为 $\pm 1/2\text{LBS}$ (即非线性误差小于 $\pm 1/2\text{LBS}$)。在做偏移量调整时,使输入电压为 $1/2\text{LBS}$ (满量程电压为 10V 时是 4.88mV),通过调整 R_1 的电阻值,使第 16 脚至第 27 脚的数字输出为 0000 0000 0000 至 0000 0000 0001 的跳变点。

③ 在图 6-7 所示的单极性输入电路原理图中,电位器 R_2 的调整端与 AD574A 芯片的第 10 脚(REF IN)连接。

④ 查阅图 6-6 可知,AD574A 芯片的第 10 脚功能的英文注释是“REFERENCE INPUT”,即基准电源电压输入端。由此可知,电位器 R_2 用于 A/D 转换电路的满量程调整。做满量程调整时,通过施加一个低于满量程值 $1\frac{1}{2}\text{LSB}$ 的模拟信号,再调整 R_2 的电阻值,使第 16 脚至第 27 脚的数字输出为 1111 1111 1110 至 1111 1111 1111 的跳变点。

【问题 3】(3 分)

这是一道要求读者结合 AD574A 工作时序理解芯片控制端标志意义的填空题。本题的解答思路如下:

① 通过阅读试题给出的 5 个供选择的答案可知,(1)~(3)空缺处需填入的是 AD574A 芯片中相应的控制端,题干中表 6-14 给出了 AD574A 工作时控制端相应的标志意义。

② 查阅表 6-14 可知,在 $\text{CE}=1$ 、 $\overline{\text{CS}}=0$ 同时满足时,AD574A 才会正常工作。在 AD574 处于工作状态时,当 $\text{R}/\overline{\text{C}}=0$ 时启动 A/D 转换,当 $\text{R}/\overline{\text{C}}=1$ 时进行数据读出操作。

③ 在试题中“如果需要 AD574A 工作于单一模式,只需将(1)、 $12/\overline{8}$ 端接至 $+5\text{V}$ 电源端,(2)和 A_0 接至 0V ,仅用 $\text{R}/\overline{\text{C}}$ 端来控制 A/D 转换的启动和数据输出。”已给出了 $12/\overline{8}$ 、 A_0 、 $\text{R}/\overline{\text{C}}$ 三个控制端,由此可排除供选择的答案中这 3 个选项。查阅表 6-14 可知,要使 AD574A 处于正常工作状态,在剩余选项中 CE 控制端需接高电平,与试题中“(1)接至 $+5\text{V}$ 电源端”相对应; $\overline{\text{CS}}$ 控制端需接低电平,与试题中“(2)接至 0V ”相对应。

④ 查阅表 6-14 可知,A/D 转换结束后,要从 AD574A 芯片的数据端读取数据,需将 $\text{R}/\overline{\text{C}}$ 控制端置 1。因此(3)空缺处可选择“ $\text{R}/\overline{\text{C}}$ ”控制端。

⑤ 另外,试题中提及的 AD574A 芯片的全速工作模式是指在 $\text{CE}=1$ 、 $\overline{\text{CS}}=0$ 同时满足时,AD574A 才会正常工作。在 AD574A 处于工作状态下,当 $\text{R}/\overline{\text{C}}=0$ 时启动 A/D 转换, $12/\overline{8}$ 和 A_0 端用来控制启动转换的方式和数据输出格式。 $\text{A}_0=0$ 时,启动转换按完整 12 位数据方式进行;当 $\text{A}_0=1$ 时,按 8 位 A/D 转换方式进行。当 $\text{R}/\overline{\text{C}}=1$ 时,AD574A 处于数据读出状态, A_0 和 $12/\overline{8}$ 控制数据输出状态的格式。当 $12/\overline{8}=1$ 时,数据以 12 位并行输出;当 $12/\overline{8}=0$ 时,数据以 8 位分两次输出。当 $\text{A}_0=0$ 时,输出转换数据的高 8 位, $\text{A}_0=1$ 时,输出 A/D 转换数据的低 4 位,这四位占一个字节的半字节,低半字节补零。

【问题 4】(2 分)

这是一道要求读者掌握 A/D 转换电路性能指标——转换时间的计算题。本题的解答思路如下:

① A/D 转换电路的转换时间是指完成一次 A/D 转换所需的时间,即由发出启动转换命令信号到转换结束信号开始有效的时间间隔。

② 通常将转换时间的倒数称为转换速率。由题干中关键信息“AD574A 芯片转换时间为 $25\mu\text{s}$ ”可计算出该芯片相对应的转换速率为 $1/25\mu\text{s}=40\text{kHz}$ 。

③ 测量某材料的性质时,以 500 点/s 的速度连续采样 1 min,对应的采样频率为 $f=500\text{ Hz}$ 。根据 A/D 电路转换时间的定义,在连续采样 1 min 的条件下,采样电路的转换时间不能超过 $\frac{1}{500 \times 60\text{ s}} = \frac{1}{30\,000\text{ s}} \approx 33.3\text{ }\mu\text{s}$ 。

④ 由于 AD574A 芯片转换时间为 $25\text{ }\mu\text{s} < 33.3\text{ }\mu\text{s}$,因此图 6-7 所示的 A/D 转换电路在转换时间指标上能满足采样要求。

【问题 5】(2 分)

这是一道要求读者掌握 A/D 转换电路性能指标——绝对精度的计算题。本题的解答思路如下:

① 在 A/D 转换中,通常将对应于一个数字量的实际模拟输入电压和理想模拟输入电压之差的最大值定义为绝对误差。

② 通常用数字量的最小有效位(LSB)的分数值来表示绝对精度。本试题中,12 位逐次逼近式 A/D 转换芯片 AD574A 的非线性误差(绝对精度)小于 $(\pm 1/2)\text{LSB}$,其最小有效位的量化单位为 9.77 mV ,因此该 A/D 转换芯片的绝对精度 $= (1/2) \times 9.77\text{ mV} = 4.885\text{ mV}$ 。

③ 另外,A/D 转换芯片的相对精度是指在整个转换范围内,任一个数字量所对应的模拟输入量的实际值与理论值之差,通常用模拟电压满量程的百分比来表示。本试题中,AD574A 转换芯片的相对精度为 0.048% 。

【问题 6】(2 分)

这是一道要求读者掌握采样保持电路功能的简答题。本题所涉及的知识点有以下几点:

① 通常 A/D 转换电路的组成框图如图 6-13 所示。

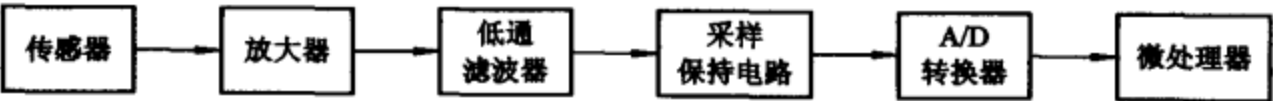


图 6-13 A/D 转换电路的组成框图

② A/D 转换电路各部件功能表如表 6-15 所示。

表 6-15 A/D 转换电路各部件功能表

部 件	主 要 任 务	自评分
传感器	把现场各种物理信号按一定规律转换成相对应的电信号	
放大器	把传感器送来的电信号放大到 A/D 转换所需要的量程范围内	
低通滤波器	选出有用的频率信号,抑制杂散的高频干扰信号,提高电路信噪比	
采样保持电路	“采样”是指跟踪输入信号;“保持”是指暂停跟踪输入信号,保持已采集的输入信号,确保在 A/D 转换期间保持输入信号不变	
A/D 转换器	把采样保持电路锁存的模拟电信号转换成相对应的数字电信号,等待微处理器用指令将其读出	
微处理器	读取 A/D 转换器中的数字信号,并按一定的算法进行数据处理,输出系统所需求的各种控制信号	

③ 由表 6-15 可知,采样保持电路主要完成两个功能,即跟踪输入的模拟电信号和锁存已采集的输入电信号,以确保在 A/D 转换期间保持输入信号不变。

试题二

【问题 1】(10 分)

这是一道要求读者掌握对分查表法算法的程序分析题。本题的解答思路如下:

① 对分查表法用于有序表的查找。对一个字节长度为 N 的有序表(从小到大排列),先取 $N/2$ 处的值与待找的数据 X 进行比较,若 X 大于 $N/2$ 处的值,则下一次取 $N/2$ 至 N 的中间值即 $3N/4$ 处的值进行比较;反之下一次取 0 至 $N/2$ 的中间值即 $N/4$ 处的值进行比较。如此不断对分查找,直到找出所需要的数据 X 。

② 结合以上算法思想,仔细分析试题的程序段。以下给出该程序段的每条语句的详细解析。

DATA	SEGMENT		;DATA 段定义伪指令
BINTAB	DB 18, 25, 32, 46, 56, 78, ...		;第 1~100 个数据
	DB 90, 95, 99, 100, 106, 126, ...		;第 101~200 个数据
	DB 189, 195, 197, 202, 206, 216, ...		;第 201~256 个数据
COUNT	EQU 256		;数据块长度
FLAG	DW ?		;查到所找数据需要查找次数的存储单元
KEY	EQU 100		;所查找的数据
DATA	ENDS		;DATA 段定义结束伪指令
STAK	SEGMENT		;STAK 段定义伪指令
STPN	DB 20 DUP(?)		
STAK	ENDS		;STAK 段定义结束伪指令
CODE	SEGMENT		;CODE 段定义伪指令
	ASSUME CS:CODE, DS:DATA, ES:DATA, SS:STAK		;段寄存器说明伪指令
BINSEAT	PROC FAR		;过程(子程序)定义伪指令
START:	MOV AX, DATA		
	MOV DS, AX		
	MOV ES, AX		
	LEA SI, BINTAB		;SI←数据表上指针
	MOV CX, COUNT		;CS←数据块长度
	MOV DX, 1		;立即数 1 送 DX, 表示第 1 次查询
	MOV AX, SI		;数据表首地址送 AX
	ADD AX, CX		;AX←数据表首地址+数据块长度
	MOV DI, AX		;DI←数据表下指针(AX)
	MOV AL, KEY		;AL←要查询的数据
LOP0:	MOV BX, SI		;数据表首地址送 BX
	ADD BX, DI		;BX←数据表首地址+数据表尾地址
	SHR BX, 1		;BX←中项指针=(SI+DI)/2, 操作数逻辑右移

			;移一位相当对其除 2
	CMP	AL, [BX]	;要查询的数据与中项数据比较
	JZ	FOUND	;结果相同, 找到则转至 FOUND
	PUSHF		;保存状态标志
	CMP	BX, SI	;比较中项指针=上指针?
	JZ	NFOUND	;相等表示未找到, 转至 NFOUND
	POPF		;恢复状态标志
	JL	LESS	;要查询的数据小于中项数据时, 转至 LESS
	MOV	SI, BX	;要查询的数据大于中项数据时, 修改上指针
	JMP	NEXT	;表示在后一半继续查找
LESS:	MOV	DI, BX	;要查询的数据小于中项数据时, 修改下指针
			;表示在前一半继续查找
NEXT:	INC	DX	;查找次数加 1
	JMP	LOP0	;重新计算中项指针继续查找
NFOUND:	MOV	AX, DX	;未找到, 标志全 1 送 DX
FOUND:	MOV	AX, DX	;查找次数送 AX
	MOV	FLAG, AX	;查找次数送 FLAG 单元
	RET		;返回主程序
BINSEAT	ENDP		;过程(子程序)定义结束伪指令
CODE	ENDS		;CODE 段定义结束伪指令
	END	START	;模块结束伪指令

③ 由以上分析可知, 根据“MOV AL, KEY”、“CMP AL, [BX]”两条语句可推理出, (1)空缺处应填入题干中要求查询的数据 126。

④ “CMP AL, [BX]”语句用于所查询的数据与中项数据比较, “JZ FOUND”语句表示如果比较结果相同, 则转至找到后的处理标号 FOUND 处, 由此可判断(2)空缺处填写的内容与所查询的数据不等于中项数据处理过程相关。

⑤ (2)空缺处的后一条语句“JZ NFOUND”语句表示如果比较结果相等, 则转至未找到处理标号 NFOUND 处, 由此可判断, 该空缺处填写的内容用于判断查找过程是否可以结束, 可通过比较中项指针是否等于数据表上指针的语句“CMP BX, SI”来完成此功能。

⑥ 由(3)空缺处所在语句的标号“LESS”可知, (3)空缺处填写的内容与“JL LESS”语句(判断所查询的数据是否小于中项数据)相关。由对分查表法算法可知, 当所查询的数据小于中项数据时, 需修改数据表下指针, 以使查找过程在前一半继续进行。因此该空缺处需填入与数据表下指针 DI 相关的语句“MOV DI, BX”。

⑦ 同理, 由(4)空缺处所在语句的标号“NEXT”可知, (4)空缺处填写的内容也与“JL LESS”语句(判断所查询的数据是否小于中项数据)相关。由对分查表法算法可知, 当所查询的数据大于中项数据时, 通过“MOV SI, BX”语句修改数据表上指针, 以使查找过程在后一半继续进行。可见该空缺处是修改数据表下指针或上指针后继续查找的公共执行语句。再由“MOV DX, 1”语句及“MOV AX, DX”、“MOV FLAG, AX”语句可推理出, (4)空缺处填写的内容是查找次数加 1 的“INC DX”语句。

⑧ 由于(5)空缺处所填写的语句是程序模块的最后一条语句, 因此由程序中启动标号“START”可推理出, 该空缺处是一条与之相对应的模块结束伪指令“END START”, 用于

告诉汇编程序源文件结束,并给出执行程序的人口位置。

⑨ 另外,为了使读者更深入掌握对分查表法的应用,下面给出用对分查表法进行子程序设计的步骤:

- a. 表的长度放在 CX 寄存器中。
- b. 将 BINTAB 表的首地址放入 SI 寄存器中。
- c. 将要搜索的关键字放在 AL 中。
- d. 计算中点元素的地址(中项指针),并放入 BX 寄存器中。
- e. 将关键字 AL 与中点元素的值进行比较,

若 $(AL) < [BX]$, 则选低值的半个表(即 SI 为首地址, $[BX]$ 为尾地址), 并转步骤 d;

若 $(AL) > [BX]$, 则选高值的半个表(即 $[BX]$ 为首地址, DI 为尾地址), 并转步骤 d;

若 $(AL) = [BX]$, 则找到并将查找次数送入 FLAG 存储单元。

【问题 2】(2 分)

这是一道要求读者掌握实现相同功能的汇编语句改写的编程题。本题的解答思路如下:

① 在汇编程序代码中“LEA SI, BINTAB”语句用于实现将数据表 BINTAB 的首地址送源变址寄存器 SI 的功能,在第 1 次查询时该地址被定义为数据表的上指针。

② 传送指令 MOV 可实现 CPU 内部寄存器之间的数据传送、寄存器与内存之间的数据传送,以及将一个立即数送给 CPU 的内部寄存器或内存单元。

③ 由于每个变量具有段属性(SEG)、偏移量属性(OFFSET)和类型属性(TYPE),其中,段和偏移量两个属性可构成变量的逻辑地址。由此与“LEA SI, BINTAB”语句所实现的功能等价的语句是:“MOV SI, OFFSET BINTAB”。

【问题 3】(3 分)

这是一道要求读者掌握汇编程序分析的综合理解题。本题的分析思路如下:

① “MOV AX, DX”、“MOV FLAG, AX”这两条语句实现将查找次数送入 FLAG 单元的功能。而题干关键信息“现要求查找其中值为 126 的数据”,在数据段定义伪指令程序段中:

```
BINTAB DB 18, 25, 32, 46, 56, 78, ...           ; 第 1~100 个数据
        DB 90, 95, 99, 100, 106, 126, ...        ; 第 101~200 个数据
        DB 189, 195, 197, 202, 206, 216, ...     ; 第 201~256 个数据
```

由该定义可知,值为 126 的数据是数据表的第 106 个数据。

② 要查询数据表中第 106 个数据,中项指针的值需经过 $\frac{1+256}{2} \approx 128$, $\frac{1+128}{2} \approx 64$, $\frac{64+128}{2} = 96$, $\frac{96+128}{2} = 112$, $\frac{96+112}{2} = 104$, $\frac{104+112}{2} = 108$ 和 $\frac{104+108}{2} = 106$ 七次变换过程才能找到所查询的数据。因此,该汇编程序执行后,FLAG 存储单元中的数值是 7,即查询到值为 126 的数据的查找次数。

③ 由于对分查表法对数据表的要求是元素按从大到小或从小到大的顺序排列,因此该算法对有序表的查找速度较快。假如数据表中有 N 个数据,则该算法最多的比较次数是 $\lg N + 1$ 。

④ 比较语句“CMP AL, $[BX]$ ”,通过比较所查询的数据与中项数据两者的值,可判断

是否已找到所要查找的数据值。

⑤ 本试题要求查询位于这 256 个字节数据表中的数据, 注意所查询的数据是数据表中的一个元素。因此“CMP AL, [BX]”语句最多被执行的次数为 $\lg N + 1 = \lg 256 + 1 = 8 + 1 = 9$ 。

试题三

【问题 1】(5 分)

这是一道要求读者根据实际应用环境分析多任务嵌入式实时操作系统的特点的综合理解题。本题的解答思路如下:

(1) 嵌入式实时系统(RTOS)是指能够在指定的时间内完成系统功能和对内部或外部、同步或异步时间做出响应的系统, 它还具有在事先定义的时间范围内识别和处理离散事件的能力, 系统能够处理和存储控制系统所需要的大量数据。实时操作系统的主要功能是多任务、可抢占的任务调度、中断管理、存储管理、时钟管理、外部设备管理以及文件系统等。

(2) 嵌入式实时操作系统的主要特点如表 6-16 所示。

表 6-16 实时操作系统的主要特点

主要特点	描 述	
多任务	不但包括周期任务、非周期任务、偶发任务, 还包括非实时任务。实时任务要求满足其时限, 而非实时任务要求其响应时间尽可能地短	
约束的复杂性	时间约束	根据截止期限, 又可分为硬实时和软实时。硬实时是指应用的时间需求应能够得到完全满足, 否则将造成重大安全事故; 软实时是指偶尔出现违反某些应用的时间需求而不会对系统运行造成严重影响
	资源约束	是指多个实时任务共享有限的资源时, 必须按照一定的资源访问控制协议进行同步, 以避免死锁等现象
	执行顺序约束	是指各任务的启动和执行必须满足一定的时间和顺序约束
	性能约束	是指必须满足可靠性、可预测性、服务质量 QoS 等性能指标
具有短暂超载能力	是指能应对系统元件出现老化、环境的动态变化、应用规模的扩大等短暂超载现象	
可预测性	是指系统能对实时任务的执行时间进行判断, 确定是否能够满足任务的时限要求	
可靠性	是指采用静态分析、保留资源、冗余配置等方法, 使系统在最坏情况下都能正常工作或避免损失	
与外部环境的交互作用性	是指实时系统必须在规定的时间内对外部环境的请求做出反应, 以及对任何异常行为采取预防措施	

(3) 由试题中关键信息“只保留了标准的多任务实时操作系统的核心功能, 例如 CPU 管理、中断管理和外设管理等”可直接理解简化的多任务嵌入式实时操作系统中“简化”一词的涵义。“简化”可加快该操作系统的开发周期, 并使其更易于维护。

(4) 由题干关键信息“对各种模拟量、开关量、脉冲量、水位落差量等数据量的监测,事实上只是周期性地重复执行相应的各个任务模块”可知,系统中有着相对较多的简单输入量,另外由于“监视的对象比较固定,系统硬件的规模相对固定不变”,因此,采用简化多任务实时操作系统是可行的。

(5) 采用简化的实时操作系统比标准多用户实时操作系统(MROS)所需占用的主存空间容量小,更符合题干中“尽可能允许把所有软件都固化在 EPROM 中”的要求。

(6) 另外,采用简化的实时操作系统具有可提高系统的实时性能和快速响应能力,可减少操作系统的运行开销(例如 CPU 可更快运行任务,更快响应中断),提高系统的资源利用率和性价比等。

【问题 2】(5 分)

这是一道要求读者根据实际应用环境分析嵌入式实时操作系统任务状态转换的综合理解题。本题的分析思路如下:

① 任务是相应任务块程序关于某数据集合的可并发的一次运行活动,即任务通常作为占有资源的最基本单位。在本案例的实时操作系统中,任务总共有五个状态,分别是休眠状态、就绪状态、运行状态、等待状态和延时等待状态。而且试题中已给定“在任何时候,一个任务只会处于其中的某一个状态”。

② 其中,休眠状态指任务还没有初始化,或任务的执行已经结束,实际上对操作系统来说,该任务是不存在的。就绪状态是指任务已经具备了运行的条件,但是由于 CPU 正忙,正在运行其他的任务,因此暂时不能运行的状态。

③ 运行状态表示此任务正在占有 CPU,并在 CPU 上运行。处于就绪状态的任务只要把 CPU 控制权分配给它(任务调度),它就能够立刻执行。

等待状态是指任务因为正在等待某种事件的发生而暂时不能运行。例如,它正在等待某个 I/O 操作的完成,或者它同某个任务之间存在着同步关系,正在等待该任务给它发信号。此时,即使 CPU 已经空闲了下来,它也还是不能运行。处于等待状态的任务可以被某些系统调用或事件激发而转到就绪状态。当一个任务运行过程中需要等待某个事件或更高优先级的任务占用 CPU 时,将由运行状态转移到等待状态。由此可判断(1)空缺处填写的内容是“等待状态”,(2)空缺处填写的内容是“就绪状态”,(5)空缺处填写的内容是“运行状态”。

④ 处于运行状态的任务如暂时不需要占用 CPU 资源,则可直接进入休眠状态;而当操作系统唤醒某一处于休眠状态的任务时,该任务将进入就绪状态,因此(3)空缺处可填入“休眠状态”等内容。

⑤ 由图 6-8 中“延时时间到”以及(2)空缺处填写的内容是“就绪状态”、(5)空缺处填写的内容是“运行状态”可推理出,(4)空缺处可填入“延时等待”这一状态。

⑥ 图 6-14 给出了一个完整的简化实时操作系统的任务状态转换图。

【问题 3】(5 分)

这是一道要求读者掌握信号量操作机制的程序分析题。本题的解答思路如下:

① 在本试题中,数据采集任务(Collect_task)和计算任务(Calculate_task)共用一个单缓冲区。数据采集任务(Collect_task)采集到一个数据后,只有当缓冲区为空时,才能将数据送入该单缓冲区存放,否则应等待单缓冲区腾空。当单缓冲区中有数据时,计算任务(Calculate_task)才能从缓冲区中取出数据进行计算,否则也应等待。

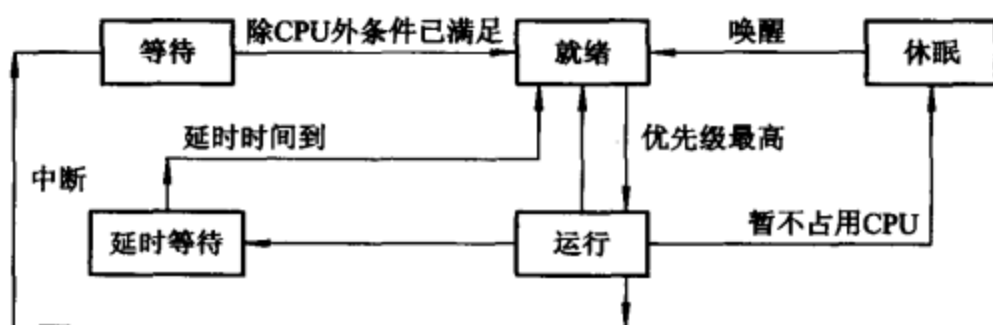


图 6-14 简化实时操作系统的任务状态转换图

② 本试题信号量操作机制的实质是一个典型的生产者——消费者操作问题，即需要使用两个任务间同步算法解决问题。

③ 本试题中应设置两个信号量，其中一个信号量来表示缓冲区中是否有可供计算的数据，其初值为 0。程序代码中“`int sf=0;`”语句正是对该信号量的定义。

④ 实现两个任务间同步算法时，还需定义另一个信号量来表示单缓冲区中是否有空位置存放新的数据，其初值为 1。数据采集任务(Collect_task)程序代码中已给出“`P(se);`”语句，间接说明另一个信号量的标识是“se”。因此(6)空缺处所填写的内容是“`se=1`”。

⑤ 当数据采集任务(Collect_task)将数据送入单缓冲区中时，应使用“`V(sf);`”语句唤醒原处于等待状态的计算任务(Calculate_task)。因此(7)空缺处所填写的内容是“`V(sf);`”语句。

⑥ 由于计算任务(Calculate_task)需要等待单缓冲区中有数据时，才能从缓冲区中取出数据进行计算，也就是说，单缓冲区中没有数据时，计算任务(Calculate_task)应处于等待状态。因此(8)空缺处实现此功能语句的语句是“`P(sf);`”。

⑦ 当计算任务(Calculate_task)从缓冲区中取出数据时，单缓冲区的空间被腾空，此时允许将新数据存入该单缓冲区，即单缓冲区的使用权限将交付给数据采集任务(Collect_task)，因此(9)空缺处需填入语句“`V(se);`”实现此功能。

⑧ 当信号量 sf 的初始值为 -1 时，假设计算任务(Calculate_task)先被调度执行，它执行 `P(sf)` 操作时，使 sf 的值变为 -2，计算任务(Calculate_task)进入阻塞状态。假设此时发生了一次时钟中断，数据采集任务(Collect_task)被调度执行，它顺利通过了 `P(se)` 操作，并使 se 的值变为 0；接着在执行 `V(sf)` 操作，并使 sf 的值变为 -1，假设满足“While(采集工作未完成)”判断条件，`P(se)` 操作将再次执行，并使 se 的值变为 -1，数据采集任务(Collect_task)也进入阻塞状态。这样一来，两个任务都在等待对方释放信号量，就成了一种死锁状态。

试题四

【问题 1】(3 分)

这是一道要求读者掌握场景法的基本流、备选流路径描述的应用分析题。本题的解答思路如下：

① 经过用例的每条不同路径都反映了基本流和备选流，都用箭头来表示。基本流用粗直黑线来表示，是经过用例的最简单的路径。图 6-9 中粗直黑线就是对该 IC 卡加油机应用系统基本流路径的描述。

② 再根据表 6-3 中 A1~A6 各个基本流的描述和描述顺序,以及图 6-9 中已给出的基本流 A1、A2、A3、A6 的路径位置可知,图 6-9 中(3)空缺处填写的内容应为“A4(输入购油量)”,(4)空缺处填写的内容应为“A5(加油)”。

③ 备选流用不同的彩色曲线表示,一个备选流可能从基本流开始,在某个特定条件下执行,然后重新加入基本流中;也可能起源于另一个备选流,或者终止用例而不再重新加入某个流。

④ 根据表 6-5 中备选流 B 的描述“在基本流 A2 过程中,该卡不能够识别或是非本机可以使用的 IC 卡,加油机退卡,并退出基本流”可知,备选流 B 的路径起源于基本流 A2,终点是基本流 A6(退还加油卡)。因此图 6-9 中(1)空缺处填写的内容就是“B(加油卡无效)”。

⑤ 同理,根据表 6-5 中备选流 C 的关键描述“在基本流 A3 过程中,判断该卡账户属于黑名单,加油机吞卡,退出基本流”可知,备选流 C 的路径起源于基本流 A3,终点是基本流 A6(退还加油卡)。因此图 6-9 中(2)空缺处填写的内容就是“C(卡账户属于黑名单)”。

⑥ 备选流 D(加油卡账面现金不足)的描述“系统判断加油卡内现金不足,重新加入基本流 A4,或选择退卡”中,“系统判断加油卡内现金不足”是指当客户输入需要购买的汽油数量时,该 IC 卡加油机应用系统将准备购买的汽油数量乘与单位油价得到需支付的费用,并将此费用与客户加油卡内的现金进行比较后的一种可能结果。因此备选流 D 的路径起源于基本流 A4(输入购油量)之后,基本流 A5(加油)之前。由备选流 D 的描述“重新加入基本流 A4,或选择退卡”可知,备选流 D 的路径终点可能是基本流 A4,以便重新进行购油量的输入,也可能是基本流 A6(退还加油卡)。因此需将 D(加油卡账面现金不足)的内容同时填入图 6-9 中(5)、(6)空缺处。

⑦ 备选流 E(加油机油量不足)的描述“系统判断加油机内油量不足,重新加入基本流 A4,或选择退卡”中,“系统判断加油机内油量不足”是指当客户输入需要购买的汽油数量时,该 IC 卡加油机应用系统将准备购买的汽油数量与系统加油机内现存的汽油数量进行比较后的一种可能结果。因此备选流 E 的路径起源于基本流 A4(输入购油量)之后,基本流 A5(加油)之前。备选流 E 的路径终点可能是基本流 A4,以便重新进行购油量的输入,也可能是基本流 A6(退还加油卡)。因此需将 E(加油机油量不足)的内容也同时填写入图 6-9 中(5)、(6)空缺处。

【问题 2】(5 分)

这是一道要求读者掌握在场景法中设计测试用例的综合分析题。本题的分析思路如下:

① 根据题干的描述可知,本案例中存在着以下五种场景。

场景 1: 基本流 A;

场景 2: 基本流 A、备选流 B;

场景 3: 基本流 A、备选流 C;

场景 4: 基本流 A、备选流 D;

场景 5: 基本流 A、备选流 E。

② 测试用例表(表 6-6)已给出了场景 1 的测试用例,对于其他 4 行所填写的内容可

以通过参照场景 1 的测试用例的解答思路进行。

③ 本案例中与场景 2(AB)相关的描述有：

A2：验证加油卡。加油机从加油卡的磁条中读取账户代码，并检查它是否属于可以接收的加油卡。

B：加油卡无效。在基本流 A2 过程中，该卡不能够识别或是为非本机可以使用的 IC 卡，加油机退卡，并退出基本流。

由“备选流 B(加油卡无效)”的描述提取出场景 2(AB)的名称——“卡无效”，输入值是“账号无效”，预期结果是“退卡”。

④ 本案例中与场景 3(AC)相关的描述有：

A2：验证加油卡。加油机从加油卡的磁条中读取账户代码，并检查它是否属于可以接收的加油卡。

A3：验证黑名单。加油机验证卡账户是否存在于黑名单中，如果属于黑名单，加油机吞卡。

C：卡账户属于黑名单。在基本流 A3 过程中，判断该卡账户属于黑名单，例如：已经挂失，加油机吞卡并退出基本流。

由“备选流 C(卡账户属于黑名单)”的描述提取出场景 3(AC)的名称——“黑名单卡”，输入值是“帐号有效”和“黑名单卡”，预期结果是“吞卡”。

⑤ 基本流 A1~A4、备选流 D 的相关描述与本案例中场景 4(AD)有关。由“备选流 D(加油卡账面现金不足)”的描述提取出场景 4(AD)的名称——“金额不足”，其输入值为“帐号有效”、“非黑名单卡”、“输入购油量有效”、“加油机油量有效”、“账面金额无效”，预期结果是“提示错误，或重新输入购油量，或退卡”。

⑥ 基本流 A1~A4、备选流 E 的相关描述与本案例中场景 5(AE)有关。由“备选流 E(加油机油量不足)”的描述提取出场景 5(AE)的名称——“油量不足”，其输入值为“帐号有效”、“非黑名单卡”、“输入购油量有效”、“账面金额有效”、“加油机油量无效”，预期结果是“提示错误，或重新输入购油量，或退卡”。

⑦ 将以上分析结果按照试题中的规定——“V 表示有效数据元素，I 表示无效数据元素，n/a 表示不适用”，归纳整理成如表 6-17 所示的测试用例表。

表 6-17 测试用例表

测试用例 ID 号	场 景	账号	是否黑名单卡	输入油量	账面金额	加油机油量	预期结果
CW01	场景 1：成功加油	V	I	V	V	V	成功加油
CW02	场景 2：卡无效	I	n/a	n/a	n/a	n/a	退卡
CW03	场景 3：黑名单卡	V	V	n/a	n/a	n/a	吞卡
CW04	场景 4：金额不足	V	I	V	I	V	提示错误，或重新输入购油量，或退卡
CW05	场景 5：油量不足	V	I	V	V	I	

【问题3】(2分)

这是一道要求读者掌握根据具体应用环境区分场景法中正面/负面测试用例的分析题。本题的解答思路如下：

① 将问题1的分析结果归纳整理为如图6-15所示的场景路径的描述图。

② 在图6-15中，对于基本流A来说，测试用例CW01一直沿着用例的基本流路径执行，未发生任何偏差，因此该测试用例是基本流A的正面测试用例。

③ 基本流A的全面测试必须包括负面测试用例，以确保只有在符合条件的情况下才执行基本流。测试用例CW02~CW05是基本流A的负面测试用例。

④ 另外说明一点，测试用例CW02~CW05相对于基本流A而言都是负面测试用例，但它们相对于各自的备选流而言是正面测试用例。对于这些备选流中的每一个来说，至少存在一个负面测试用例——测试用例CW01。

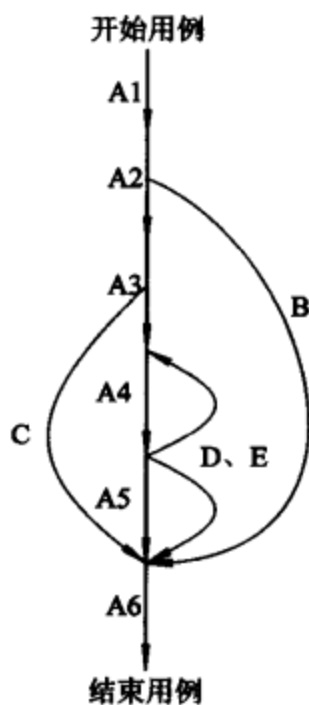


图6-15 完整的场景路径的描述图

【问题4】(2分)

这是一道要求读者说明测试用例矩阵优点的简答题。本题所涉及的知识点有：

① 用表6-6所示的方法创建测试用例矩阵时，无需为条件输入任何实际的值，只需查看表中的V和I项就能判断是否已经确定了充足的测试用例。即用表6-6所示的方法创建测试用例矩阵的优点在于容易看到测试的是什么条件。

② 从表6-17中可以看出，数据元素“输入油量”仅有“V”项，缺少“I”项（例如输入“-1”等情况），这表明表6-17的测试用例还不完整。

【问题5】(3分)

这是一道要求读者应用边界值法设计测试用例的分析题。本题的解答思路如下：

① 边界值分析法是一种黑盒测试方法，是对等价类分析方法的一种补充。选择其测试用例取值的原则有：如果输入条件规定了值的范围，则应该取刚达到这个范围的边界值，以及刚刚超过这个范围边界的值作为测试输入数据；如果输入条件规定了值的个数，则用最大个数、最小个数、比最大个数多1个以及比最小个数少1个的数做为测试数据；如果程序的规格说明给出的输入域或输出域是有序集合（如有序表、顺序文件等），则应选取集合的第一个和最后一个元素作为测试用例；如果程序用了一个内部结构，则应该选取这个内部数据结构的边界值作为测试用例，分析规格说明，找出其他可能的边界条件。

② 由“油价为5元/升，用户的账户金额为800元”可计算出用户合理的购油量输入范围是0~160升。

③ 本案例边界值法的取值原则可按照取最小值、最大值、比最小值稍小的值和比最大值稍大的值进行。

④ 因此，用边界值法设计基本流A4输入油量的测试数据是最小值0升，（由于比0小的数是负数，不符合现实意义，因此不取），最大值160升，比最大值稍大的161升。

试题五

【问题 1】(3 分):

这是一道要求读者掌握 CAN 总线传输媒体和逻辑电平的基本常识题。本题的解答思路如下:

① 由题干关键信息“系统主控机与网络节点采用双绞线连接”可知,双绞线是 CAN 总线常用的传输媒体之一。因此(1)空缺处可填入“双绞线”等内容。

② 由题干关键信息“CAN_H 比 CAN_L 高,表示逻辑‘0’”可知,在数字电路中与逻辑“0”信号相对应的是逻辑“1”信号,再者“显性”与“隐性”也是一对相对的词语,因此可推理出(2)空缺处需填入内容是“逻辑‘1’”。

③ 当“显性”位和“隐性”位同时发送时,最后总线数据将为“显性”。正是这一特性为 CAN 总线的仲裁奠定了基础。因此(3)空缺处可填入“显性”或“逻辑 0”等内容。

【问题 2】(3 分)

这是一道要求读者根据 CAN 总线的传输特性、帧的结构来估算数据帧等待传输时间的分析计算题。本题的解答思路如下:

① 由题干关键信息“图 6-10 所示的网络结构中有 12 个网络节点”和“系统主控机与网络节点采用双绞线连接,实验时最大节点间的距离小于 35 m”,而试题中已给出“当 CAN 信号传输距离在 40 m 以内时,其最高通信速率为 1 Mb/s”,而且“网络上节点的数目主要取决于总线驱动电路,目前可达 110 个节点”等信息间接可推理出,在数据帧等待传输时间估算过程中数据传输速率可取 1 Mb/s。

② 试题中关键信息“每帧内有 CRC 校验及其他校验措施(适用于位数小于 127 位的帧)”可推理出,在数据帧等待传输时间估算过程中数据帧长度可取 127 位。再加上传输的标识或起止位,则最小数据帧长为 134 位。

③ CAN 最高通信速率为 1 Mb/s,传送 1 bit 用时 $1\ \mu\text{s}$,那么传送一帧 134 位的数据需用时 $134\ \mu\text{s}$ 。

④ 因为 CAN 总线网络上节点信息允许分成不同的优先级,从而可以满足不同的实时需求。“具有最高优先权的数据”是指该数据帧在最坏情况下获得的传输权限。由于 CAN 总线刚开始传输一个帧,就要把这个帧传完,才能使下一个最高优先级的数据帧获得传输的机会。

⑤ 由以上分析可知,具有最高优先权的数据帧最快可在 $134\ \mu\text{s}$ 时间内获得传输权限(CAN2.0A 的规范)。

【问题 3】(4 分)

这是一道要求读者掌握 CAN 总线接口电路作用的理解分析题。本题的解答思路如下:

① 在图 6-11 所示的电路图中,MCP2510 芯片组成的 CAN 控制器主要完成时序逻辑转换等工作,负责 CAN 通信协议的执行,承担网络通信控制任务。

② 通常嵌入式微处理器电路的 I/O 电平是 TTL 电平,而 CAN 控制器采用的是 CAN 总线电平,要在电气特性上满足 CAN 总线标准,还需要一个转换芯片(通常称之为 CAN 总线的物理层芯片),用它来实现 TTL 电平到 CAN 总线电平特性的转换。图 6-11 所示的电路图中 TJA1050 芯片所组成的电路正是 CAN 总线收发器,它是 CAN 控制器与物理

总线之间的接口电路。

【问题 4】(2 分)

这是一道要求读者掌握电平转换 I/O 接口电路元器件参数选择的分析题。本题的解答思路如下:

① 在图 6-11 所示的电路图中, MCP2510 芯片组成的 CAN 控制器使用的是 3.3 V 电压供电, 而 TJA1050 芯片组成的 CAN 总线收发器使用的是 5.0 V 电压供电。

② 试题中给出的关键信息“MCP2510 和 TJA1050 连接的两个信号都是单向信号”间接告知, 在图 6-11 所示的电路图中, 对于 MCP2510 来说, TXCAN 是输出信号, RXCAN 是输入信号, 只需要单向满足 I/O 接口的电气特性即可。

③ 试题中已给出在 3.3 V 电压供电情况下, MCP2510 芯片输入信号即 RXCAN 高电平的范围 V_{ih} 是 2~4.3 V。这并不能满足 5 V 逻辑的 TJA1050 芯片的输出电平, 因此需要进行 I/O 接口电路的电平转换。在图 6-11 所示的电路图中, 采用电阻分压(电阻 SR10、SR11)的方法实现单向的电平转换。

④ 在图 6-11 所示的电路图中, 分压电阻 SR10、SR11 的阻值选择需要考虑 TJA1050 芯片输出信号的驱动能力和 MCP2510 的 RXCAN 管脚的输入阻抗两个关键因素。TJA1050 芯片最多可以输出 15 mA 的电流, 这足以驱动电阻 $RS11 + RS10 = 10\text{ k}\Omega + 20\text{ k}\Omega = 30\text{ k}\Omega$ 。MCP2510 芯片的输入电流 I_{LI} 不会超过 5 μA 。

⑤ 另外说明一点, 在图 6-11 所示的电路图中, TJA1050 为 5 V 供电时, 输入高电平 V_{ih} 范围是 2~5.3 V。而 3.3 V 供电的 MCP2510 输出的 TXCAN 信号高电平 V_{oh} 的最小值为 2.6 V, 因此 MCP2510 输出的 TXCAN 信号电平可以满足 TJA1050 的输入电平的要求。

【问题 5】(3 分)

这是一道要求读者掌握 CAN 网络中作为现场采集的网络节点软件功能模块设计的综合分析题。本题的解答思路如下:

① 仔细阅读试题信息可知, 在图 6-10 所示的网络结构中, 每一个网络节点中至少包括了接口初始化功能模块。

② 由题干中关键信息“图 6-10 所示的网络结构中有 12 个网络节点, 每一节点都通过传感器采集现场的有关数据”可知, 现场采集的网络节点的首要功能就是数据采集。因此, 每一个网络节点中必须有一个“数据采集”功能模块。

③ 由题干中关键信息“系统主控机……可以向网络节点发送命令”可推理出, 每一个网络节点中必须有一个与之相对应的“参数、命令接收模块”。同理由“系统主控机……接受节点数据”可知, 每一个网络节点必须根据系统主控的命令, 主动地把采集到的数据、节点的工作状态以及相关的响应信息等经过 CAN 总线传输给系统主控机。因此, 每一个网络节点中必须有一个“主动发送”功能模块。

④ 通常一个嵌入式 CAN 网络节点必须支持 CAN2.0B 协议规定——标准 CAN 总线数据帧格式和扩展数据帧格式。在数据传输之前, 每一个网络节点必须先对采集到的数据, 或者内部的状态信息作一些初步的处理, 从而保证符合 CAN 总线中帧格式的要求。因此, 在每一个网络节点中必须有一个数据帧预处理模块。

6.1.4 参考答案

1. 上午试题参考答案

表 6-18 给出本模拟试卷问题(1)至问题(75)的参考答案,供读者练习时参考,以便查漏补缺。读者可按每空 1 分的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

表 6-18 参考答案及评分标准

题 号	参考答案	题 号	参考答案
(1)~(5)	B、B、C、C、B	(41)~(45)	A、A、A、B、A
(6)~(10)	C、B、C、D、C	(46)~(50)	C、D、C、A、B
(11)~(15)	C、C、A、D、D	(51)~(55)	C、D、A、B、A
(16)~(20)	B、A、B、C、D	(56)~(60)	A、A、D、B、D
(21)~(25)	A、B、B、C、B	(61)~(65)	D、B、C、C、A
(26)~(30)	B、B、B、C、A	(66)~(70)	B、C、A、B、D
(31)~(35)	C、B、B、B、B	(71)~(75)	B、C、A、D、A
(36)~(40)	A、B、A、B、B		

2. 下午试题参考答案

表 6-19 给出本模拟试卷试题一至试题五的参考答案,供读者练习时参考,以便查漏补缺。读者也可依照所给出的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

表 6-19 参考答案及评分标准

试题	问题与分值	参考答案及评分标准	自评分
一	【问题 1】 (2 分)	0~10 V 的输入接在第 13 脚和第 9 脚(1 分) 0~20 V 的输入接在第 14 脚和第 9 脚(1 分)	
	【问题 2】 (4 分)	电位器 R1 用于 A/D 转换电路的偏移量调整(2 分) 电位器 R2 用于 A/D 转换电路的满量程调整(2 分)	
	【问题 3】 (3 分)	(1) A 或 CE (2) D 或 \overline{CS} (3) C 或 R/\overline{C} (每空 1 分)	
	【问题 4】 (2 分)	以 500 点/s 的速度连续采样 1 min, 对应的采样频率为 $f=500\text{ Hz}$ (1 分)。在连续采样 1 min 的条件下, 采样电路的转换时间不能超过 $\frac{1}{500 \times 60\text{ s}} = \frac{1}{30\,000\text{ s}} \approx 33.3\text{ }\mu\text{s}$ 。由于 AD574A 芯片转换时间为 $25\text{ }\mu\text{s}$ $< 33.3\text{ }\mu\text{s}$, 因此该 A/D 转换电路在转换时间指标上能满足采样要求 (1 分)	
	【问题 5】 (2 分)	由于 AD574A 的非线性误差小于 $(\pm 1/2)\text{ LBS}$, 其最小有效位的量化单位为 9.77 mV , 因此该 A/D 转换芯片的绝对精度 $= (1/2) \times 9.77\text{ mV} \approx 4.885\text{ mV}$ (2 分)	
	【问题 6】 (2 分)	跟踪输入的模拟电信号(1 分); 锁存已采集的输入电信号, 以确保在 A/D 转换期间保持输入信号不变(1 分)	

续表

试题	问题与分值	参考答案及评分标准	自评分
二	【问题 1】 (10 分)	(1) 126, 或其对应的二进制数形式 (3) MOV DI, BX (5) END START(每空 2 分)	
	【问题 2】 (2 分)	MOV SI, OFFSET BINTAB(2 分)	
	【问题 3】 (3 分)	FLAG 存储单元中的数值是 7(或其对应的二进制数形式)(2 分) CMP AL, [BX]语句最多被执行的次数为 9 次(1 分)	
三	【问题 1】 (5 分)	① 可加快该操作系统的开发周期, 并使其更易于维护; ② 减少操作系统占有内存的容量, 便于固化; ③ 减少操作系统的运行开销(例如 CPU 占用时间); ④ 可提高系统的实时性能和快速响应能力; ⑤ 可提高系统的资源利用率和性价比等(每小点 1 分)	
	【问题 2】 (5 分)	(1) 等待状态 (3) 休眠状态 (5) 运行状态(每空 1 分)	
	【问题 3】 (5 分)	(6) se=1 (8) P(sf); (10) 死锁(每空 1 分, 注意(7)~(9)语句的“;”)	
四	【问题 1】 (3 分)	(1) B (4) A5	
	【问题 2】 (5 分)	(2) C (5) D、E	
	【问题 3】 (2 分)	(3) A4 (6) D、E(每空 0.5 分)	
	【问题 4】 (2 分)	参见本案例要点解析中的表 6-17	
	【问题 5】 (3 分)	正面测试用例 CW01(1 分) 负面测试用例 CW02~CW05(1 分)	
	【问题 4】 (2 分)	容易看到测试的是什么条件(1 分), 无需为条件输入任何实际的值, 只需查看表中的 V 和 I 项就能判断是否已经确定了充足的测试用例(1 分)	
	【问题 5】 (3 分)	0 升、160 升、161 升(每个数据 1 分)	

续表

试题	问题与分值	参考答案及评分标准	自评分
五	【问题 1】 (3 分)	(1) 双绞线 (2) 逻辑“1” (3) “显性”或“逻辑 0” (每空 1 分)	
	【问题 2】 (3 分)	由于仅有 12 个网络节点且最大节点间的距离小于 35 m, 故数据传输速率取 1 Mb/s(1 分)。数据帧长度取 127 位, 再加上传输的标识或起止位, 则最小数据帧长为 134 位(1 分)。具有最高优先权的数据帧最快可在 134 μ s 时间内获得传输权限(1 分)	
	【问题 3】 (4 分)	MCP2510 芯片组成的 CAN 控制器主要完成时序逻辑转换等工作, 负责 CAN 通信协议的执行, 承担网络通信控制任务(2 分) TJA1050 芯片所组成的电路正是 CAN 总线收发器, 它是 CAN 控制器与物理总线之间的接口电路, 实现 TTL 电平到 CAN 总线电平特性的转换(2 分)	
	【问题 4】 (2 分)	需要考虑 TJA1050 芯片输出信号的驱动能力(1 分)和 MCP2510 的 RXCAN 管脚的输入阻抗等两个关键因素(1 分)	
	【问题 5】 (3 分)	数据采集功能模块(1 分) 数据信息发送功能、命令参数接收模块(1 分) 内部数据预处理功能模块(1 分)	

6.2 模拟试题 2

6.2.1 上午试题

(考试时间 9:00—11:30 共 150 分钟)

请按下述要求正确填写答题卡。

(1) 在答题卡的指定位置上正确写入你的姓名和准考证号, 并用正规 2B 铅笔在你写入的准考证号下填涂准考证号。

(2) 本试卷的试题中共有 75 个空格, 需要全部解答, 每个空格 1 分, 满分 75 分。

(3) 每个空格对应一个序号, 有 A、B、C、D 四个选项, 请选择一个最恰当的选项作为答案, 在答题卡相应序号下填涂该选项。

(4) 解答前请务必阅读例题和答题卡上的例题填涂样式及填涂注意事项。解答时用正规 2B 铅笔正确填涂选项, 如需修改, 请用橡皮擦干净, 否则会导致不能正确评分。

例题

• 2007年下半年全国计算机技术与软件专业技术资格(水平)考试日期是(88)月(89)日。

(88) A. 9 B. 10 C. 11 D. 12

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

因考试日期是“11月3日”，故(88)选C，(89)选C，应在答题卡序号(88)下对C填涂，在序号(89)下对C填涂(参看答题卡)。

试题

• 在计算机系统中，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，根据Amdahl定律，这样做可以使整个系统的性能提高(1)倍。

(1) A. 1.6 B. 1.7 C. 1.8 D. 1.9

• 利用高速通信网络将多台高性能工作站或微型机互连构成机群系统，其系统结构形式属于(2)计算机。

(2) A. SISD B. MISD C. SIMD D. MIMD

• 文件系统采用多重索引结构搜索文件内容。设块长为512字节，每个块号长3字节，如果不考虑逻辑块号在物理块中所占的位置，那么三级索引时可寻址的文件最大长度为(3)。

(3) A. 3.46 MB B. 7.05 MB C. 599.73 MB D. 1199.46 MB

• 新项目与过去成功开发过的项目类似，但规模更大，这时应该使用(4)进行项目开发设计。

(4) A. 原型法 B. 变换模型 C. 瀑布模型 D. 螺旋模型

• 在面向对象分析过程中，用概念模型来详细描述系统的问题域，用(5)来表示概念模型。(6)关系用于表示类与类、接口与接口之间的继承关系；在Java中，用(7)关键字来直接表示这种关系。

(5) A. 序列图

B. 构件图

C. 协作图

D. 类图

(6) A. 实例关系

B. 泛化关系

C. 聚合关系

D. 依赖关系

(7) A. extends

B. implements

C. generalization

D. realization

• 在《GB/T19001—1994 在计算机软件开发、供应、安装和维护中的使用指南》(idt ISO9000—3:1997)中，“idt”是一种(8)关系。

(8) A. 采用

B. 等同采用

C. 等效采用

D. 非等效采用

• 甲企业开发出某一新嵌入式路由器产品，并投入生产。乙企业在甲企业之后三个月也开发出同样的嵌入式路由器产品，并向专利部门提交专利申请。在乙企业提交专利权申

请后的第 5 日, 甲企业向该专利部门提交了与乙企业相同的专利申请。按照专利法的有关条款, (9) 将获得专利申请权。

- (9) A. 甲乙企业同时 B. 乙企业
C. 甲乙企业先后 D. 甲企业

• 在图 6-16 所示的系统中, 员工小郭登录 Internet 网时需要三层交换机、防火墙和路由器三台设备(每种设备赋予相同的可靠度)均正常运行。若要求整个系统可靠度为 0.98, 那么每一种设备的可靠度至少为(10)。

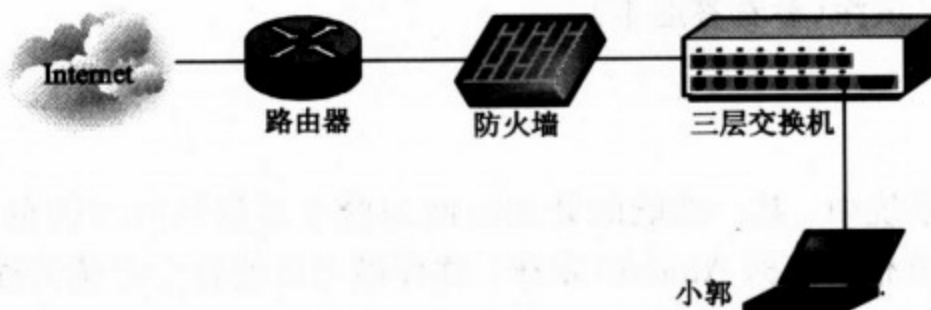


图 6-16 Internet 接入拓扑结构图

- (10) A. 0.27 B. 0.73
C. 0.98 D. 0.99

• 对 20~4000 Hz 频率段的声音信号进行脉码调制编码, 若采样量化级为 128 级, 则应选择数据传输率至少达到(11)的 Modem。

- (11) A. 64 kb/s B. 56 kb/s C. 48 kb/s D. 32 kb/s

• 蓝牙技术工作在全球通用的 2.4GHz ISM 频段, 其最高数据速率为(12)。

- (12) A. 1 Mb/s B. 11 Mb/s C. 54 Mb/s D. 100 Mb/s

• 当前磁盘读写位于柱面号 20, 此时有多个磁盘请求, 以下列柱面号顺序送至磁盘驱动器 10、22、20、2、40、6、38。寻道时, 移动一个柱面需 6 ms, 采用改进电梯调度算法所需寻道时间为(13)。

- (13) A. 348 ms B. 360 ms C. 376 ms D. 876 ms

• RAID 级别是指磁盘阵列中硬盘的组合方式, 不同级别的 RAID 为用户提供的磁盘阵列在性能上和安全性的表现上都有所不同。以下(14)是具有磁盘镜像和双工功能的磁盘阵列。

- (14) A. RAID 1 B. RAID 5 C. RAID 7 D. RAID 10

• 为了保障数据的存储和传输安全, 需要对一些重要数据进行加密。由于对称密码算法(15), 因此特别适合对大量的数据进行加密。国际数据加密算法 IDEA 的密钥长度是(16)位。

- (15) A. 比非对称密码算法更安全 B. 比非对称密码算法密钥长度更长
C. 能同时用于身份认证 D. 比非对称密码算法效率更高

- (16) A. 56 B. 64 C. 128 D. 256

• 根据统计显示, 80% 的网络攻击源于内部网络, 因此必须加强对内部网络的安全控制和防护。下面的措施中, 无助于提高同一局域网内安全性的措施是(17)。

- (17) A. 使用防火墙防止内部攻击 B. 使用日志审计系统

C. 使用入侵检测系统

D. 使用防病毒软件

- 最适合在 ADSL 接入网上实时传输视频数据的 MPEG 系列标准是(18)。

(18) A. MPEG-2 B. MPEG-4 C. MPEG-7 D. MPEG-21

- 一幅分辨率为 $800 \times 600 \times 256$ 色未压缩图像的数据量最小约为(19)KB。

(19) A. 470 B. 480 C. 4800 D. 120 000

- 在系统转换的过程中,旧系统和新系统一起工作一段时间,再由新系统代替旧系统的转换策略称为(20)。

(20) A. 直接转换 B. 间接转换 C. 分段转换 D. 并行转换

- $F(A, B, C, D) = \sum m(5) + \sum d(1, 7, 11, 13, 15)$ 的最简与或表达式是(21)

(21) A. $F = \overline{A}BCD$ B. $F = \overline{A}CD$ C. $F = \overline{A}BD$ D. $F = BD$

- 施密特触发器常应用于对脉冲波形(22)的场合。

(22) A. 定时 B. 计数 C. 整形 D. 产生

- 表达式 $251 - 53 = 176$ 在(23)进制时成立。

(23) A. 六 B. 七 C. 八 D. 十六

- PCB 设计不当,会对电子设备的可靠性产生不良影响。以下关于 PCB 地线设计方法描述错误的是(24)。

(24) A. 只由数字电路组成的 PCB 的地线系统时,可将接地线做成闭环路

B. 电路板上既有高速逻辑电路,又有线性电路时,应尽量加大线性电路的接地面积

C. 接地线应尽可能加粗,使它能通过 3 倍于 PCB 的允许电流

D. 在低频电路中,通常信号的工作频率小于 1 MHz,它的布线和元件间的电感影响较小,而接地电路形成的环流对干扰影响较大,因而应采用就近多点接地

- IP 核是实现片上系统(SOC)的基本构件,完成结构描述的 IP 核模块是(25)。

(25) A. Soft IP Core

B. Hard IP Core

C. Firm IP Core

D. DSP Core

- 以下关于嵌入式系统的组成说法错误的是(26)。

(26) A. 在一片嵌入式处理器基础上添加电源电路、时钟电路和存储器电路,就构成了一个嵌入式核心控制模块,其中嵌入式操作系统和应用程序都可以固化在 RAM 中

B. Cache 位于主存和嵌入式微处理器内核之间,用于减小主存或辅助存储器对微处理器内核造成的存储器访问瓶颈,使处理速度更快,实时性更强

C. 板级支持包(BSP)位于嵌入式系统硬件层和系统软件层之间,也称为硬件抽象层(HAL),它将系统上层软件与底层硬件分离开来,使系统的底层驱动程序与硬件无关

D. 系统软件层位于嵌入式中间层和应用软件层之间,由 RTOS、文件系统、GUI、网络系统及通用组件模块等组成。

- 嵌入式系统的结构组成中,板级支持包(BSP)也称为硬件抽象层(HAL),其特点是(27)。

- (27) A. 硬件无关性, 操作系统无关性
B. 硬件相关性, 操作系统无关性
C. 硬件无关性, 操作系统相关性
D. 硬件相关性, 操作系统相关性

• 在实时系统中, 响应时间是衡量系统性能的最重要因素。任务调度的目的是缩短系统平均响应时间, 提高系统资源利用率。以下关于嵌入式系统实时调度技术说法错误的是(28)。

- (28) A. 抢占式调度具有实时性好, 调度算法较简单, 可保证高优先级任务的时间约束, 上下文切换少等优点
B. 非抢占式调度是指不允许任务在执行期间被中断, 任务一旦占用嵌入式微处理器就必须执行完毕或自愿放弃
C. 静态表驱动策略是一种离线调度策略, 运行时刻表一旦生成后就不再发生变化
D. 在任务运行过程中, 采用静态优先级调度的任务优先级不会发生变化

• 为了进行差错控制, 必须对传送的数据帧进行校验。CRC-16 标准规定的生成多项式为 $G(x) = X^{16} + X^{15} + X^2 + 1$, 它产生的校验码是(29)位。如果 CRC 的生成多项式为 $G(X) = X^4 + X + 1$, 信息码字为 10110, 则计算出的 CRC 校验码是(30)。

- (29) A. 2 B. 15 C. 16 D. 33
(30) A. 0100 B. 1010 C. 0111 D. 1111

• 如果信息长度为 5 位, 要求纠正 1 位错, 则按照海明编码, 需要增加的最少的校验位是(31)。

- (31) A. 3 B. 4 C. 5 D. 6

• 在 MCS-51 系统中, 指令 MOVX A, @DPTR 的寻址方式称为(32)

- (32) A. 寄存器寻址 B. 寄存器间接寻址
C. 直接寻址 D. 立即数寻址

• Intel 公司用于网络处理的 IXP 系统微处理器支持 DDR 内存接口。在 200 MHz 外频条件下, DDR 内存带宽可达(33)。

- (33) A. 0.2 Gb/s B. 1.6 Gb/s C. 2.1 Gb/s D. 3.2 Gb/s

• 以下几种 A/D 转换器中转换速率最快的是(34)。

- (34) A. 并联比较型 B. 逐次渐近型
C. 计数型 D. 双积分型

• D/A 转换器通常使用(35)指标来描述其速度。

- (35) A. 转换速率 B. 线性度
C. 建立时间 D. 切换速度

• 阅读以下程序段, 在实模式存储管理方案中, gvCh 存放在(36)中; main 函数编译后的代码存入在(37)中; 指针 p 存放在(38)中。

```
#include <malloc.h>
unsigned char gvCh;
unsigned short gvShort;
```

```
unsigned int gvInt=0x12345678;
unsigned long gvLong=0x23456789;
void main(void)
{ unsigned char array[10], *p;
  p=malloc(10 * sizeof(char));
  while(1);
}
```

- (36) A. .text 段 B. .data 段 C. .bss 段 D. 堆空间
(37) A. .text 段 B. .data 段 C. .bss 段 D. 栈空间
(38) A. 栈空间 B. .data 段 C. .bss 段 D. 堆空间

• 在实现文件系统时,为加快文件目录的检索速度,可利用“文件控制块分解法”。假设目录文件存放在磁盘上,每个盘块 512 字节。文件控制块占 64 字节,其中文件名占 8 字节,文件内部号和文件其他描述信息占 56 字节。设某一目录文件共有 254 个文件控制块,采用分解法前查找该目录文件的某一个文件控制块的平均访问磁盘次数为(39)。若目录文件分解前占用 n 个盘块,则分解后改用 m 个盘块存放文件名和文件内部号部分,那么访问磁盘次数减少的条件是(40)。

- (39) A. 2 B. 4 C. 8 D. 16
(40) A. $m < n-1$ B. $m < n-2$ C. $m > n-1$ D. $m > n-2$

• RS-232-C 是目前常见的一种接口标准,它是由 EIA 组织制定的。该标准的设计数据速率是(41)b/s。

- (41) A. 6400 B. 9600 C. 19 200 D. 20 000

• 从编程的角度看,ARM 微处理器的工作状态一般有两种。(42)状态下处理器执行 32 位的、字对齐的指令

- (42) A. ARM B. Cold fire C. Thumb D. MIPS

• 以下关于 I²C 总线说法正确的是(43)。

- (43) A. SDA 由低电平向高电平跳变, SCL 为低电平时,结束传送数据
B. SDA 由低电平向高电平跳变, SCL 为高电平时,开始传送数据
C. SCL 为高电平, SDA 由高电平向低电平跳变时,开始传送数据
D. SCL 为高电平, SDA 由低电平向高电平跳变时,结束传送数据

• 以下哪种措施无助于降低嵌入式系统的功耗设计(44)。

- (44) A. 设计外部中断唤醒电路 B. 使用 UPS 稳压电源
C. 采用 DVS 技术 D. 动态调整微处理器的时钟频率和电压

• 利用 ADSL G. lite 的最高速率标准,传送一个 100 MB 的文件需要的最短时间约(45)秒。

- (45) A. 582.5 B. 694.5 C. 1638.4 D. 6553.6

• ICMP 报文封装在(46)协议数据单元中传送,在网络中起着差错和拥塞控制的作用。常用的 ping 程序中使用了回送请求/应答报文,以探测目标主机是否可以到达。

- (46) A. IP B. TCP C. UDP D. PPP

• 通常嵌入式系统加电后运行的第一段软件代码是(47)。

- (53) A. 6186 B. 12 330
C. 34 858 D. 因产生缺页中断,暂时无法获得

• 嵌入式系统支持的内存块为大页时,可分成大小为(54)的子页。

- (54) A. 1 KB B. 4 KB
C. 16 KB D. 64 KB

• 嵌入式系统对应用软件的开发和编程语言的选择提出了不同的要求。以下所考虑的一些因素说法错误的是(55)。

- (55) A. 通常嵌入式系统要求具有实时处理的能力,这种实时性主要是靠硬件层来体现的
B. 通常嵌入式系统要求支持多任务,能够处理并发事件
C. 嵌入式软件的开发需要交叉编译工具
D. 电源能量的供应在嵌入式系统中是一项约束条件

• 以下正确描述嵌入式程序代码生成过程的是(56)。

- (56) A. 高级语言代码→目标代码→汇编语言代码→二进制代码
B. 目标代码→高级语言代码→汇编语言代码→二进制代码
C. 高级语言代码→汇编语言代码→目标代码→二进制代码
D. 高级语言代码→汇编语言代码→二进制代码→目标代码

• 以下关于程序运行时内存分配区域的描述中,说法错误的是(57)。

- (57) A. 全局变量区域,存放了程序当中的所有全局变量,是静态分配的区域
B. 主函数和其他函数的代码在一个程序开始运行时,就被装入到内存,保存在代码区
C. 堆是在函数调用时分配的,当函数调用结束之后,相应的堆就会被释放
D. 栈区域是由系统自动分配的

• 当一次函数调用发生时,对其执行过程描述正确的是(58)。

- (58) A. 分配一个栈帧→复制实参变量的值→控制流转移到该函数的起始位置→开始执行该函数→控制流返回到函数调用点
B. 控制流转移到该函数的起始位置→复制实参变量的值→分配一个栈帧→开始执行该函数→控制流返回到函数调用点
C. 控制流转移到该函数的起始位置→分配一个栈帧→复制实参变量的值→开始执行该函数→控制流返回到函数调用点
D. 复制实参变量的值→控制流转移到该函数的起始位置→分配一个栈帧→开始执行该函数→控制流返回到函数调用点

• 以下关于C语言的一些叙述,不正确的是(59)。

- (59) A. 在调用函数时,实参和对应形参在类型上只需赋值兼容
B. 函数形参可以说明为 register 变量
C. 函数中的自动变量可以赋初值,每调用一次,赋一次初值
D. 外部变量的隐含类别是自动存储类别

• 典型的干扰传播路径是通过导线的传导和空间的辐射。以下无助于切断干扰传播

路径的措施是(60)。

- (60) A. 嵌入式系统电源尽可能采用磁珠和电容组成 π 形滤波电路
B. 在嵌入式微处理器的信号输入接口电路加隔离 π 形滤波电路
C. 用地线把晶振时钟区隔离起来, 并将晶振外壳接地
D. 嵌入式微处理器和大功率器件要单独接地

• 在嵌入式系统软件的开发过程中往往采取各种措施来提高软件的易维护性。如采用良好的编程风格有助于提高软件的易理解性; (61)有助于提高软件的易修改性。

- (61) A. 高效的算法
B. 信息隐蔽原则
C. 增强健壮性
D. 身份认证

• 下列叙述中, 与提高软件可移植性相关的是(62)。

- (62) A. 选择时间效率高的算法
B. 尽可能减少注释
C. 选择空间效率高的算法
D. 尽量用高级语言编写系统中对效率要求不高的部分

• 高级程序设计语言中用于描述程序中的运算步骤、控制结构及数据传输的是(63)。

- (63) A. 语句 B. 语义 C. 语用 D. 语法

• 硬件是嵌入式系统运行的载体, 也是嵌入式系统的基础。嵌入式系统硬件的选择包括硬件平台和嵌入式处理器的选择、外围设备的选择和接口电路的选择。以下关于硬件平台的选择说法错误的是(64)。

- (64) A. 如果这是个全新的项目, 最好选用能够满足需求的多个处理器来实现
B. 在多处理器设计中, 把控制和管理用一个处理器实现
C. 可以从简单的处理器入手, 随着系统负载的增加而增加处理器的数量
D. 如果已有的系统实现了相似的功能, 应尽可能重用该系统结构

• 以下关于嵌入式系统的软件设计的叙述中, 说法错误的是(65)。

- (65) A. 把应用程序转换成可以在目标机上运行的二进制代码通常需要经过编译、链接和定址等三个步骤
B. 编译过程生成的文件就是可以在嵌入式平台上执行的二进制文件
C. 编译过程产生的所有目标文件被链接成一个目标文件
D. 定址过程会把物理存储器地址指定给目标文件的每个相对偏移处

• Prior to the availability of enterprise EDM, locating a document over a LAN could be difficult, and over a WAN (66) impossible. With the model for collaborative work spreading, and with more users sharing (67) through E-mail and the Internet, enterprise EDM had become a necessity.

Many businesses are finding their greatest competitive advantage is the ability to harness the document (68). EDM systems manage this lifecycle from creation, revision, storage and (69) to routing and workflow. They also allow users to work on documents collaboratively.

Reusability of information is EDM's main benefit, as well as the key to creating the paradigms that drive business process (70).

- | | | | |
|-------------------|---------------|----------------|------------------|
| (66) A. personal | B. available | C. structural | D. nearly |
| (67) A. format | B. documents | C. text | D. network |
| (68) A. database | B. controller | C. lifecycle | D. center |
| (69) A. retrieval | B. route | C. path | D. save |
| (70) A. messages | B. packets | C. information | D. reengineering |

• Although the bulk of industry resources and energies have focused on developing the fastest (71) or slickest (72), more and more mindshare is turning to the evolution of the computer interface. Advancements in the areas of input devices, (73) processing and virtual reality could lead to fundamental changes in the way human and computer interact. The technological battlefield of the future will be adding layers between the user and the raw machine to make the (74) as invisible as possible. (75) represents the next evolutionary step for the interface.

- | | |
|------------------------------|-------------------------|
| (71) A. microprocessor | B. format |
| C. structure | D. procedure |
| (72) A. DBMS | B. application software |
| C. operating systems | D. GUI |
| (73) A. explicit | B. voice |
| C. implicit | D. special |
| (74) A. inconsistent | B. independent |
| C. interface | D. incompatible |
| (75) A. application software | B. eye-tracking device |
| C. application programs | D. virtual reality |

6.2.2 下午试题

试题一

阅读以下关于嵌入式系统 I²C 总线接口电路的技术说明, 根据要求回答问题 1 至问题 4。(15 分)

【说明】

I²C 总线是一种多主控器总线, 对多个 IC 器件具有控制总线的能力。采用串行 I²C 总线技术设计接口电路, 可节约嵌入式系统微处理器的 I/O 端口。图 6-17 给出了一种实现了键盘、时钟和实时参数存储三个功能模块的应用电路。图 6-17 中仅应用 I²C 总线构成单主控器的系统。此时时钟线 SCL 仅由微处理器驱动, 因此可以用微处理器的一根 I/O 线作为 SCL 信号线, 将其设置为输出方式, 由软件控制产生串行时钟信号; 用微处理器的另一根 I/O 线作为 I²C 总线的串行数据线, 由软件控制在时钟的低电平期间读取或输出数据。

在图 6-17 中采用 ATMEL 公司的 E²PROM AT24C64(IC8)实现了系统初始化数据、仪表系数、用户设置的参数或某些重要的实时数据参数的存储。DS1302 时钟芯片可以对年、月、日、周日、时、分、秒进行计时, 且具有闰年补偿功能, 工作电压宽达 2.5~

5.5 V。8 位并行 I/O 口扩展器件 PCF8574 具有口输出锁存功能，可直接驱动大电流 LED 管，同时还带有一个中断请求输出端 INT(漏极开路方式)。在图 6-17 中，将 PCF8574 器件的 8 位准双向口外挂了一个 4×4 键盘矩阵。

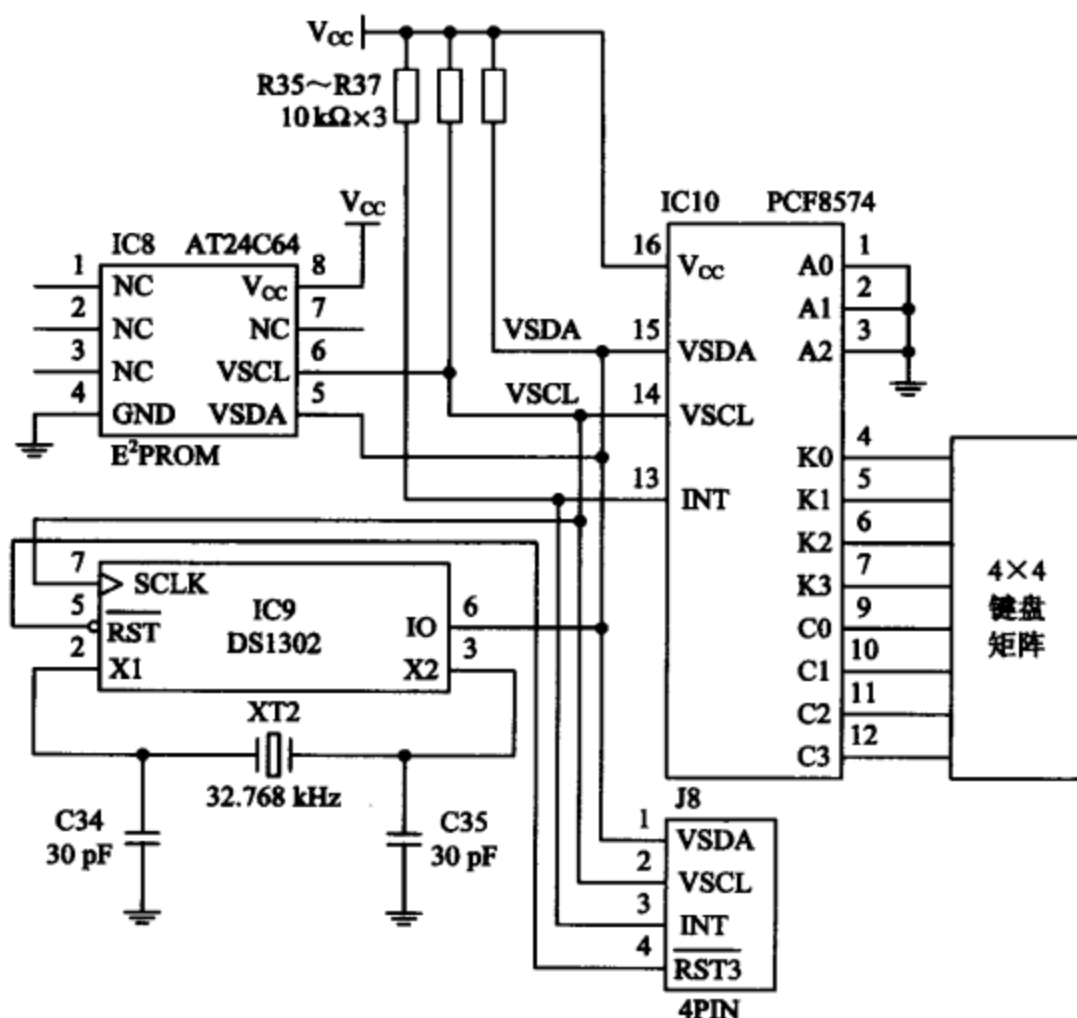


图 6-17 I²C 总线接口电路图

【问题 1】(2 分)

电阻 R36、R37 在图 6-17 电路中有何作用。请在 100 字以内简要说明。

【问题 2】(4 分)

图 6-18 给出了 I²C 总线开始条件与停止条件的时序图。通常 I²C 总线的开始条件、停止条件总是由(1)产生。在开始条件产生后，总线被认为处于(2)，在完成数据传输产生停止条件后，总线被认为处于空闲状态。在时钟信号的高电平期间，SDA 传输线上的电平必须稳定，只有在 SCL 传输线上的时钟信号为(3)，数据线上的高低电平才允许发生变化，此时传送的数据才有效。每一个写到 SDA 传输线上的数据字节必须为 8 位长度，每一次传输的字节数没有限制，每传输一个字节必须跟一个(4)，传输字节时最高位在前。

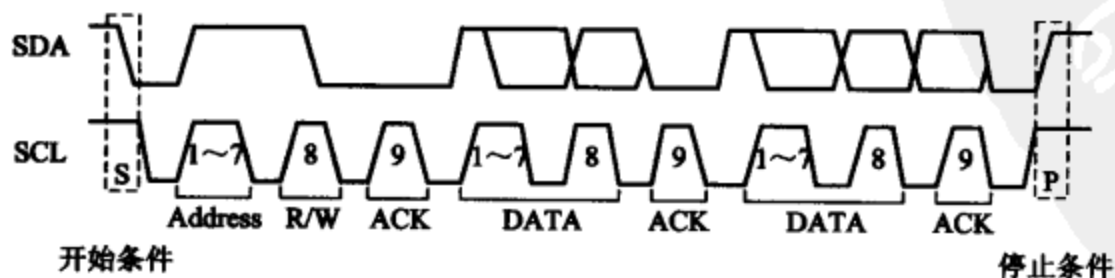


图 6-18 I²C 总线开始条件与停止条件的时序图

【问题 3】(4 分)

I²C 总线的寻址过程是在(5)后发送第 1 个字节,该字节地址决定主控器选择哪一个从器件,还可以通过(6)方式同时寻址所有的 I²C 器件,但当某个 I²C 器件不需要进行数据传输时,将忽略该寻址方式而不作任何应答。如果某个 I²C 器件需要获取数据,则将发出(7)信号并作为一个(8)。

【问题 4】(5 分)

以下是利用 MCS-51 系列单片机汇编语言实现模拟 I²C 总线接收 N 个字节数据的程序段(RDNBYTE),请将汇编程序代码中(9)~(13)空缺处的内容填写完整。表 6-21 所示为 MCS-51 系列单片机部分汇编指令及寄存器说明。

表 6-21 MCS-51 系列单片机部分汇编指令及寄存器说明

指令	功 能	寄存器	功 能
MOV	数据传送指令	A	累加器
INC	加 1 指令	R1	寄存器
JB	转移指令,直接寻址位为 1 时转移	R3	寄存器
DJNZ	转移指令,寄存器/直接寻址单元减 1,结果不为零时转移	F0	用户标志位
SJMP	相对短转移		
LCALL	长调用指令		
RET	子程序返回指令		

程序中已定义了标号为 START 的启动位子程序、标号为 STOP 的停止位子程序、标号为 MACK 的发送应答位子程序、标号为 MNACK 的发送非应答位子程序、标号为 CACK 的应答位检查子程序、标号为 WRBYT 的向 VSDA 线上发送 1 个数据字节的子程序、标号为 RDBYT 的从 VSDA 线上读取 1 个数据字节的子程序。调用 RDNBYTE 程序的归一化操作命令是

```
MOV SLA, #SLAR
MOV NUMBYTE, #N
LCALL RDNBYT
```

【汇编程序代码】

```
VSDA      EQU    P1.6      ; 虚拟 I2C 总线数据线
VSCL      EQU    P1.5      ; 虚拟 I2C 总线时钟线
SLA       EQU    50H       ; 寻址字节存放单元
NUMBYTE   EQU    51H       ; 传送字节数存放单元
MTD       EQU    30H       ; 发送数据缓冲区
MRD       EQU    40H       ; 接收数据缓冲区
SLAW0     EQU    0A0H      ; 写 AT24C64(器件地址为 0A0H)
SLAR0     EQU    0A1H      ; 读 24C64
SLAW1     EQU    40H       ; 写 PCF8574(器件地址为 040H)
SLAR1     EQU    41H       ; 读 PCF8574
```


; 模拟 I²C 总线接收 N 个字节数据

RDNBYTE: MOV R3, NUMBYTE

LCALL START ;发送启动位

MOV A, SLA ;发送寻址字节(读)

(9)

(10)

JB F0, RDNBYTE

RDN: MOV R1, #MRD

RDN1: (11)

MOV @R1, A

DJNZ R3, ACK ;N 个字节读完否? 未完转

(12)

LCALL STOP ;发送停止位

RET

ACK: (13)

INC R1

SJMP RDN1 ;转读下一个字节数据

(9)~(13)空缺处供选择的答案

A. LCALL START

B. LCALL MACK

C. LCALL CACK

D. LCALL RDBYT

E. LCALL WRBYT

F. LCALL MNACK

G. LCALL STOP

试题二

阅读以下关于基于嵌入式系统的住宅安全系统的技术说明, 根据要求回答问题 1 至问题 4。(15 分)

【说明】

基于某嵌入式系统的住宅安全系统可使用传感器(如红外探头和摄像头等)来检测各种意外情况, 如非法进入、火警及水灾等。

房主可以在安装该系统时配置安全监控设备(如传感器、显示器和报警器等), 也可以在系统运行时修改配置, 通过录像机和电视机监控与系统连接的所有传感器, 并通过控制面板上的键盘与系统进行信息交互。在安装过程中, 系统给每个传感器赋予一个 ID 编号和类型, 并设置房主密码以启动和关闭系统, 设置传感器事件发生时自动拨出的电话号码。当系统检测到一个传感器事件时, 就激活警报, 拨出预置的电话号码, 并报告关于位置和检测到的事件的性质等信息。图 6-19 所示是住宅安全系统的顶层数据流图, 图 6-20 所示是住宅安全系统的第 0 层数据流图, 图 6-21 所示是对住宅安全系统的第 0 层数据流图中加工 4 的细化图。

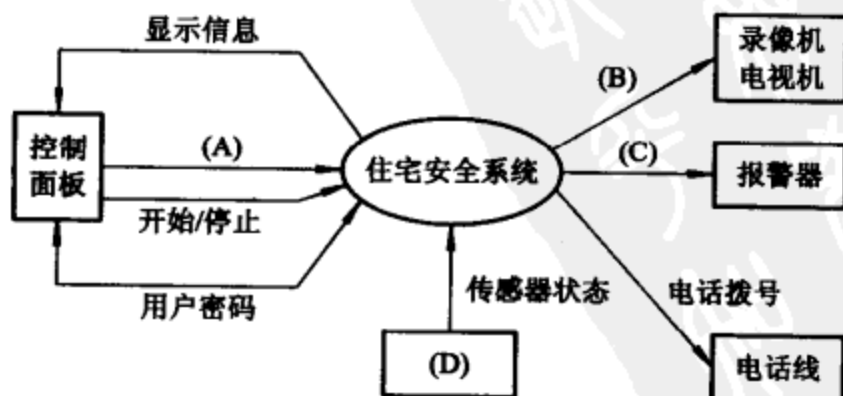


图 6-19 住宅安全系统的顶层数据流图

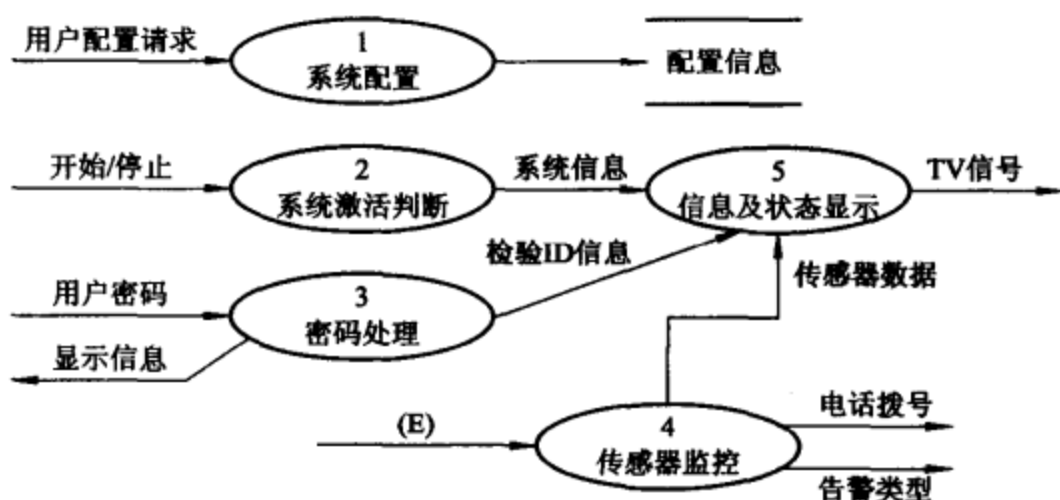


图 6-20 住宅安全系统第 0 层数据流图

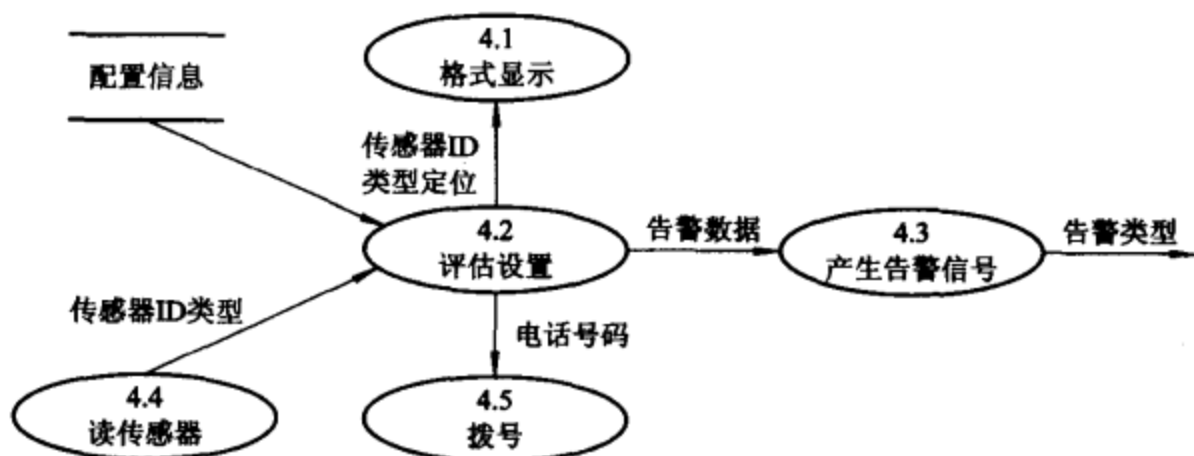


图 6-21 加工 4 的细化图

【问题 1】(5 分)

请将住宅安全系统的顶层数据流图(图 6-19)中(A)~(D)空缺处,以及第 0 层数据流图(图 6-20)中(E)空缺处的内容填写完整。

【问题 2】(3 分)

将图 6-21 所示的加工 4 的细化图中的数据流补充完整,请按以下答题格式分别指明加工名称、数据流名称和数据流方向(输入/输出)。

答题格式示例:

[4.2 评估设置]→(告警数据)→[4.3 产生告警信号]

或[4.3 产生告警信号]→(告警类型)

【问题 3】(3 分)

修改在住宅安全系统第 0 层数据流图(图 6-20)中的数据存储“配置信息”,将会影响第 0 层数据流图中的哪些加工?

【问题 4】(4 分)

数据流图是系统分析阶段用于描述系统逻辑模型的图形描述工具。嵌入式实时系统分析阶段的主要任务是确定需要解决的问题或需要完成的目标及其(1),同时对实时系统的软/硬件做全面的分析,并对软/硬件做合理的分解,为实时系统的设计打下基础。实时系统的分析需要建模和(2),以便系统分析人员估计“时间和大小”。建立系统模型时应明确体现(3)、(4)、功能特点及约束条件等因素。

试题三

阅读以下说明和 x86 汇编语言代码，根据要求回答问题 1 至问题 3。(15 分)

【说明】

在某嵌入式安全监测系统中，对某任务的加工操作需通过数据采集(Collect_task)、计算(Calculate_task)这两个不同的程序段来完成，并且执行程序段 Collect_task 和 Calculate_task 的顺序及次数有如下约定：

Collect_task(2 次)→Calculate_task(1 次)→Collect_task(5 次)→Calculate_task(4 次)
→Collect_task(2 次)→Calculate_task(2 次)

表 6-22 所示为 x86 系统部分指令及寄存器说明。

表 6-22 x86 系统部分指令及寄存器说明表

指令	功 能	寄存器		功 能
MOV	传送字或字节	段寄存器	CS	代码段寄存器
LEA	地址传送		DS	数据段寄存器
ADD	加法指令		SS	堆栈段寄存器
SUB	减法指令		ES	附加段寄存器
PUSH	进栈操作	数据寄存器	AX	累加器
POP	出栈操作		BX	基址寄存器
SAL	算术左移		CX	计数寄存器
CALL	子程序调用		DX	数据寄存器
DEC	减 1 指令	专用寄存器	BP	基数指针寄存器
JMP	无条件转移指令		SP	堆栈指针寄存器
JZ/JNZ	结果为 0/不为 0 时转移		SI	源变址寄存器
JL/JG	结果小于/大于转移		DI	目的变址寄存器
JC/JNC	有进(借)位/无进(借)位时转移			
INT	软中断			
RET	返回主程序			
IRET	从中断服务程序返回			

采用逻辑尺控制法实现以上要求的汇编程序如下：

【汇编程序代码】

```
N EQU (1)
RULE EQU (2)

CODE SEGMENT
ASSUME CS: CODE
START: MOV AX, RULE
```

```

MOV    CL, N
LOP:   SAL    AX, 1
        JC     (3)
Collect: CALL Collect_task      ;执行 Collect_task 程序段
        JMP    (4)
Calculate: CALL Calculate_task  ;执行 Calculate_task 程序段
NEXT:   (5)
        JNZ    (6)
        MOV    AH, 4CH          ;功能号送入 AH 寄存器
        INT    (7)              ;结束程序运行, 中断返回
        (8)
END START

```

【问题 1】(10 分)

请根据试题的要求, 将汇编程序代码中(1)~(8)空缺处的内容填写完整。

【问题 2】(3 分)

汇编程序是一种系统软件, 它的基本功能是将汇编语言源程序翻译成(1)。对于一个 N 行汇编语言代码的源程序, 由于汇编指令中形成操作数地址的部分可能出现后面才会定义的符号, 因此汇编程序一般至少需要(2)次扫描程序才能完成翻译过程。

【问题 3】(2 分)

编译程序的功能是什么? 解释程序与编译程序的主要区别是什么? 请在 150 字以内简要说明。

试题四

阅读以下说明和 Socket 程序, 根据要求将程序代码中(1)~(10)空缺处的内容填写完整。(15 分)

【说明】

网络应用的基本模型是客户机/服务器模型, 这是一个不对称的编程模型, 通信的双方扮演不同的角色, 分别是客户机和服务器。

一般发起通信请求的应用程序称为客户软件, 该应用程序通过与服务器进程建立连接来发送请求, 然后等待服务器返回所请求的内容。服务器软件一般是指等待接收并处理客户机请求的应用程序, 通常由系统执行, 等待客户机请求, 并且在接收到请求之后, 根据请求的内容, 向客户机返回合适内容。

本题中的程序较为简单, 客户机接收用户在键盘上输入的文字内容, 服务器将客户机发送来的文字内容直接返回给客户机, 在通信过程中服务器方和客户机方都遵守的通信协议如下:

由客户机首先发送请求, 该请求由首部和内容两大部分组成, 两个部分各占一行文字, 通过行结束符'\n'隔离。

首部只有一个 Length 域, 用于指定请求的内容部分的长度, 首部的结构为 '关键词 Length'+' '+数值+' \n'

内容部分为一行文字, 其长度必须与 Length 域的数值相符。

例如, 客户机的请求为“Length 14\n hello, welcome to my home!”, 服务器接收请求处理后返回文字“Hello, welcome to my home!”。

【Socket 程序】

```
//服务器主程序部分
#include<stdio.h>
.....
#define SERVER_PORT 8080 //引用头文件部分略
#define BACKLOG 5 //服务器监听端口号为 8080
//连接请求队列长度
int main(int argc, char * argv[ ])
{
    int listenfd, connfd; //监听套接字、连接套接字描述符
    struct sockaddr_in servaddr; //服务器监听地址
    listenfd = (1); //创建用于监听的套接字
    if(listenfd<0)
    {
        fprintf(stderr, "创建套接字错误!"); //套接字创建失败时打印错误信息
        exit(1);
    }
    bzero(&servaddr, sizeof(servaddr)); //将地址结构置空
    servaddr.sin_family = AF_INET; //设置地址结构遵循 TCP/IP 协议
    servaddr.sin_addr.s_addr = htonl(2);
    //设置监听的 IP 地址为任意合法地址, 并将该地址转换为网络字节顺序
    servaddr.sin_port = (3); //设置监听的端口, 并转化为网络字节顺序
    if(bind((4))<0)
    {
        fprintf(stderr, "绑定套接字与地址!");
        exit(1);
    } //将监听地址与用于监听的套接字绑定, 绑定失败时打印错误信息
    if(listen(listenfd, BACKLOG)<0)
    {
        fprintf(stderr, "转换套接字为监听套接字!");
        exit(1);
    } //将用于监听的套接字由普通套接字转化为监听套接字
    for(;;)
    {
        connfd = (5);
        //从监听套接字的连接队列中接收已经完成的连接并创建新的连接套接字
        if(connfd<0)
        {
            fprintf(stderr, "接收连接失败!");
            exit(1);
        } //接收失败打印错误信息
```

```

        serv_respon(connfd);    //运行服务器的处理函数
        (6);                    //关闭连接套接字
    }
    close(listenfd);            //关闭监听套接字
}

//服务器通信部分
#include<stdio.h>
.....                          //引用头文件部分略
void serv_respon(int sockfd)
{
    int nbytes;
    char buf[1024];
    for(;;)
    {
        nbytes=read_requ(sockfd, buf, 1024);
                                //读出客户机发出的请求,并分析其中的协议结构,获知请求的内
                                //容部分的长度,并将内容复制到缓冲区 buf 中
        if(nbytes == 0)
            return;              //如客户机结束发送就退出
        else if(bytes<0)
        {
            fprintf(stderr, "读错误信息: %s\n", strerror(errno));
            return;
        } //读请求错误打印错误信息
        if(write_all(sockfd, buf, nbytes)<0)
            //将请求中的内容部分反向发送回客户机
            fprintf(stderr, "写错误信息: %s\n", strerror(errno));
    }
}

int read_requ(int sockfd, char * buf int size)
{
    char inbuf[256];
    int n;
    int i;
    i = read_line(sockfd, inbuf, 256);
                                //从套接字接收缓冲区中读出一行数据,该数据为客户请求的首部
    if(i<0)
        return(i);
    else if(i==0)
        return(0);
    if(strncmp(inbuf, " ", 6) == 0)

```



```

        sscanf((7), "%d", &n); //从缓冲区 buf 中读出长度信息
    else
    {
        sprintf(buf, " ", 14);
        return(14);
    } //取出首部 Length 域中的数值, 该数值为内容部分的长度
    return(read_all(sockfd, buf, n)); //从接收缓冲区中读出请求的内容部分
}

```

```
int get_char(int fd, char * ch)
```

/* get_char 的处理方式较为特殊, 并不是每次调用 read 函数读一个字符, 而是一次从缓冲区中读一块内容, 再一次一个字符提交给函数 read_line, 如果提交完了就再读一块, 这样可以提高读缓冲区的效率。另外, 由于客户机是分两次调用 writ_all 函数将请求的首部和内容发送给服务器, 因此 get_char 不会取出请求内容部分的字符 */

```

{ //声明静态变量, 在 get_char 多次被调用期间, 该变量的内存不释放
    static int offset=0;
    static int size=0;
    static char buff[1024];
    for( ; size <= 0 || (8); )
    {
        size = read(fd, buff, 1024); //一次从套接字缓冲区中读出一个数据块
        if(size<0)
        {
            if(errno==EINTR)
            {
                size=0;
                continue; //EINTR 表示本次读操作没有成功, 但可以继续使用套接字读出数 i
            }
        }
        else
        {
            return(-1);
        }
    }
    offset = 0; //读出数据后, 将偏址置为 0
    *ch = buff[(9)]; //将当前的字符取出, 并将偏址移向下一字符
    return(1);
}

```

```
int read_line (int fd, char * buf, int maxlen)
```

//函数 read_line 的作用是读出请求的首部, 其处理的方法是每次调用 get_char 函数
//取出一个字符, 检查该字符是否是回车符'\n', 如果是回车符, 就返回请求的首部

```

{
    int i, n;
    char ch;
    for(i=0; i<maxlen; )
    {

```

```

        n = get_char(fd, &ch); //取出一个字符
        if(n == 1)
        {   buff[i++] = ch; //将字符加入字符串中
            if((10))
                break;
        }
        else if(n < 0)
            return(-1);
        else
            break;
    }
    buf[i] = '\0';
    return(i);
}

```

【部分 Socket 数据结构与函数说明】

1. 地址结构

① sockaddr_in 类型的结构定义：

sockaddr_in 是通用套接字结构 sockaddr 在 TCP/IP 协议下的结构重定义，为 TCP/IP 套接字地址结构。

```

Struct sockaddr_in{
    short int sin_family;           //地址类型 AF_XXX, 其中 AF_INET 为 TCP/IP 专用
    unsigned short int sin_port;    //端口号
    struct in_addr sin_addr;        //Internet 地址
    //端口号以及 Internet 地址使用的是网络字节顺序, 需要通过函数 htons 转换
}

struct in_addr {
    _u32 s_addr;                   //类型为 unsigned long
}

```

② hostent 类型的结构定义：

```

struct hostent{
    char * h_name;                  //主机的正式名称
    char * * h_aliases;             //别名列表
    int h_addrtype;                 //主机地址类型: AF_XXX
    int h_length;                   //主机地址长度: 4 字节(32 位)
    char * * h_addr_list;           //主机 IP 地址列表
}

#define h_addr h_addr_list[0]

```

2. 基本函数

① int socket(int domain, int type, int protocol);

函数 socket 创建一个套接字描述符，如果失败则返回 -1。domain 为地址类型；type

为套接字类型, 本题中为 SOCK_STREAM; protocol 指定协议, 本题中为 0。

② int connect(int sockfd, struct sockaddr * servaddr, int addrlen);

函数 connect 与服务器建立一个连接, 成功返回 0, 失败返回 -1。servaddr 为远程服务器的套接字地址, 包括服务器的 IP 地址和端口号; addrlen 为地址的长度。

③ int read(int fd, char * buf, int len);

int write(int fd, char * buf, int len);

函数 read 和 write 从套接字读和写数据, 成功返回数据量大小, 否则返回 -1。buf 指定数据缓冲区, len 指定接收或发送的数据量大小。

④ int bind(int sockfd, struct sockaddr * myaddr, int addrlen);

函数 bind 将本地地址与套接字绑定在一起, 成功返回 0, 否则返回 -1; myaddr 是本地地址; addrlen 为套接字地址结构的长度。

⑤ int listen(int sockfd, int backlog);

函数 listen 将一个套接字转换为倾听套接字, 成功返回 0, 否则返回 -1; backlog 为请求队列的最大长度。

⑥ int accept(int sockfd, struct sockaddr * addr, int * addrlen);

函数 accept 从监听套接字的完成连接中接收一个连接, 如果完成连接队列为空, 那么这个进程睡眠, 失败则返回 -1, 成功时返回新的套接字描述符。sockfd 为监听套接字, addr 为客户机的地址, addlen 为地址长度, 在调用时用常量 NULL 代替 addr 与 addlen 表示无需取出客户机的地址信息。

⑦ struct hostent * gethostbyname(const char * hostname);

函数 gethostbyname 查询指定的域名地址对应的 IP 地址, 返回一个 hostent 结构的指针, 如果不成功则返回 NULL。

3. 用户自定义函数

① int read_all(int fd, void * buf, int nbyte);

函数 read_all 从参数 fd 指定的套接字描述符中读取 nbytes 字节数据至缓冲区 buf 中, 成功则返回实际读的字节数(可能小于 nbyte), 失败则返回 -1。

② int write_all(int fd, void * buf, int nbyte);

函数 write_all 向参数 fd 指定的套接字描述符中写入缓冲区 buf 前 nbyte 字节的数据, 成功则返回实际写的字节数(始终等于 nbyte), 失败则返回 -1。

③ write_requ 函数为客户机发送请求的函数。

④ read_requ 函数为服务器获取请求的函数。

试题五

阅读以下关于嵌入式操作系统软件编码优化的技术说明, 根据要求回答问题 1 至问题 5。(15 分)

【说明】

由于嵌入式系统对实时性的要求较高, 因此一般要求对代码的性能进行优化, 使代码的执行速度越快越好。图 6-22 给出了两个算术运算的程序段(a)、(b)。

<pre> struct{ int a; char b; int c; }foo[10]; int i; for(i=0; i<10; i++) { foo[i].a=33; foo[i].b=55; foo[i].c=77; } </pre> <p style="text-align: center;">(a)</p>	<pre> struct{ int a; char b; int c; }*fp, *fend, foo[10]; fend=foo+10; for(fp=foo; fp!=fend; ++fp) { fp->a=33; fp->b=55; fp->c=77; } </pre> <p style="text-align: center;">(b)</p>
--	---

图 6-22 src2.c 文件

【问题 1】(2 分)

在嵌入式系统设计过程中,给定一份软件设计规格说明书后,下一步的工作就是编写代码。通常编码工作包含哪些步骤?

【问题 2】(4 分)

图 6-22 所示的(a)、(b)程序段的功能是完全一样的,都是对一个结构体数组的各个元素进行初始化,但采用两种不同的方法来实现。请在 200 字以内归纳这两个算术运算程序段所采用的实现方法。

【问题 3】(4 分)

在一台采用 S3C44B0X 微处理器(32 位 RISC 结构)的嵌入式系统中,将图 6-22 所示的(a)、(b)程序段分别重复 11 000 次,(a)程序段需要 2.13 ms;(b)程序段需要 1.01 ms。

由此可见,在进行算术运算编码时遵守哪些编码准则?

【问题 4】(3 分)

在嵌入式系统中,由于软/硬件资源有限,且系统对实时性和可靠性要求较高,因此在进行嵌入式软件开发时,要注意对执行时间、存储空间和开发/维护时间这三种资源的使用进行优化。请具体说明在编写代码时,需要做到哪几点编码准则?

【问题 5】(2 分)

编译器对程序进行一定的优化是非常有必要的,高级语言的程序可以被转化为汇编语言的指令形式。通常优化的方法有(1)、循环优化、废代码的清除等。

对嵌入式系统进行性能测试可以衡量系统的实时性,分析嵌入式软件的优劣,同时分析程序的执行时间可以帮助分析 CPU 功耗等特性。最坏执行时间是指(2)。

6.2.3 要点解析**1. 上午试题要点解析**

(1) C。在计算机系统中,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,根据 Amdahl 定律计算公式:系统加速比 $S_p = \frac{1}{(1-f) + \frac{f}{r}}$ 。依

题意将 $f=0.5$, $r=10$ 代入计算, 可得 $S_p=1.8$ 。

(2) D。利用高速通信网络将多台高性能工作站或微型机互连构成机群系统, 这种机群系统适用于中、粗粒度的并行进程的高效并行处理。很显然, 它属于多指令流多数据流 MIMD 的系统结构。而阵列处理机是单指令流多数据流 SIMD 系统结构的典例, 它适用于矩阵运算。

(3) D。文件系统采用多重索引结构搜索文件内容。设块长为 512 字节, 每个块号长 3 字节, 那么一个索引块可以存放 170 个盘块号。

如果不考虑逻辑块号在物理块中所占的位置, 且文件系统采用二级索引时, 那么最多可包含存放文件的盘块号总数 $N=170 \times 170=28\,900$ 个盘块, 可寻址的文件的最大长度 $L=28\,900 \times 256\text{ B}=7225\text{ KB}=7.05\text{ MB}$ 。

如果文件系统采用三级索引, 那么最多可包含存放文件的盘块号总数 $N=170 \times 170 \times 170=4\,913\,000$ 个盘块; 可寻址的文件的最大长度 $=4\,913\,000 \times 256\text{ B}=1\,228\,250\text{ KB}=1199.46\text{ MB}$ 。

(4) C。因为新项目比过去成功开发过的某项目规模更大, 且有成功的项目开发经验可以借鉴, 所以该项目开发设计模型适宜采用瀑布模型(即生命周期开发模型)。

(5) D; (6) B; (7) A。在面向对象分析过程中, 用概念模型来详细描述系统的问题域。类图用于描述系统的结构化设计, 即用来表示概念模型, 它最基本的元素是类或接口, 表达了类、接口以及它们之间的静态结构和关系。表 6-23 给出了类图的六种关系。

表 6-23 类图的六种关系

序号	关 系	说 明	箭头方向	关键字
1	泛化(Generalization)	表示类与类、接口与接口之间的继承关系	由子类指向父类	extends
2	实例(Realization)	用于指定两个实体之间的一个合同	由实现接口的类指向被实现的接口	implements
3	关联(Association)	表示类与类之间的连接, 即一个类保存对另一个类实例的引用, 并在需要的时候调用这个实例的方法	单向箭头指向遍历或查询的方向; 而双向箭头是可选的	
4	聚合(Aggregation)	是关联的一种形式, 代表两个类之间的整体/局部关系		
5	合成(Composition)	是聚合的一种特殊形式, 暗示“局部”在“整体”内部的生存期职责		
6	依赖(Dependency)	也是类与类之间的连接, 并且依赖总是单向的, 是类间最弱的一种关系		

另外提醒一点: 聚合暗示着整体在概念上处于比局部更高的一个级别, 在实例图中不存在回路, 即只能是一种单向关系; 而关联暗示两个类在概念上位于相同的级别。在 Java 中, 关联使用实例变量来实现; 聚合也是使用实例变量来实现的。关联和聚合的区别纯粹

是概念上的,在Java语法上没有刻意的区分。

(8) B。我国标准与国际标准的对应关系主要有4种管理方法,分别是:①等同采用(代号idt);②等效采用(代号eqv);③非等效采用(代号neq);④修改采用(代号mod)。

(9) B。遵照“谁先申请谁拥有,同时申请则协商归属,而不同时驳回”的专利申请规则,本案例中,虽然甲企业的新路由器产品比乙企业早开发出来,并且投入了生产,但是其提交专利权申请的日期比乙企业迟了5天,因此按照专利法有关条款,乙企业将获得专利申请权。

(10) D。在该网络拓扑中,员工小郭要上Internet需要具有相同可靠度的三层交换机、防火墙和路由器三台设备均正常运行。如果整个系统的可靠度 $R=0.98$,那么每一种设备的可靠度 $R_i \geq \sqrt[3]{R} = \sqrt[3]{0.98} = 0.9933$ 。

(11) B。对204 000 Hz频率段的声音信号进行脉码调制PCM编码,利用数据传输速率基本公式 $S=B \lg N$ 计算,该声音信号经128级量化后的数据传输速率为

$$S=B \lg N = f_s \times \lg N = 8000 \times \lg 128 = 56 \text{ kb/s}$$

其中采样频率 f_s 为8000 Hz,是声音信号最高频率4000 Hz的2倍。

(12) A。蓝牙技术工作在全球通用的2.4 GHz ISM频段,采用跳频扩频技术和时分多路利用技术使特定移动电话、便携式电脑和各种便携式通信设备的主机之间在近距离内(通常为10 m以内)实现无缝的资源共享,其最高数据速率为1 Mb/s。

(13) D。当前磁盘读写位于柱面号20,寻道时移动一个柱面需6 ms。当10、22、20、2、40、6、38柱面号顺序送至磁盘驱动器时,①按改进的电梯调度算法计算。磁头移动顺序为(20)→20→22→38→40→10→6→2,磁头移动总量是58柱面,总寻道时间为 $58 \times 6 \text{ ms} = 348 \text{ ms}$ 。

②先来先服务调度算法计算。磁头移动顺序为(20)→10→22→20→2→40→6→38,磁头移动总量是146柱面,总寻道时间是: $146 \times 6 \text{ ms} = 876 \text{ ms}$ 。

③下一个最邻近柱面调度算法计算。磁头移动顺序为(20)→20→22→10→6→2→38→40,磁头移动总量是60柱面,总寻道时间是: $60 \times 6 \text{ ms} = 360 \text{ ms}$ 。

(14) A。RAID级别是指磁盘阵列中硬盘的组合方式,不同级别的RAID为用户提供的磁盘阵列在性能上和安全性的表现上都有所不同。RAID 1磁盘阵列具有磁盘镜像和磁盘双工功能。RAID 5的工作方式是将各个磁盘生成的数据校验切成块,分别存放到组成阵列的各个磁盘中去,这样就缓解了校验数据存放时所产生的瓶颈问题,但是分割数据及控制存放都要付出速度上的代价。RAID 7是目前最高档次的磁盘阵列。而RAID 10是RAID 0和RAID 1的功能组合。

(15) D; (16) C。为了保障数据的存储和传输安全,需要对一些重要数据进行加密。由于对称密码算法比非对称密码算法效率更高,因此特别适合对大量的数据进行加密。国际数据加密算法IDEA的密钥长度是128位。

(17) A。根据统计显示,80%的网络攻击源于内部网络,因此,可以使用日志审计系统、入侵检测系统、网络防病毒软件等措施来加强对内部网络的安全控制和防范。防火墙技术主要用于防止外网的恶意攻击和限制内网用户访问外网某些非法站点,但不能用它来防止内网的恶意攻击。

(18) B。在MPEG系列标准中,MPEG-4最适合在公用电话交换网(PSTN)和ADSL

接入网上实时传输视频数据。MPEG-2 为数字电视标准；MPEG-7 为多媒体内容描述接口标准；MPEG-21 为多媒体框架结构标准。

(19) A。当像素颜色等级有 $256=2^8$ 色，且不采用压缩算法时，存储一个像素需要 8 位，即占用 1 个字节存储空间。一张照片的像素数为 800×600 像素，则存储该幅图像所需的存储空间为 $800\times600\times1\text{ B}=480\,000\text{ B}=468.75\text{ KB}\approx470\text{ KB}$ 。

(20) D。新系统试运行成功之后，就可以在新系统和旧系统之间互相转换。新旧系统之间的转换方式通常有直接转换、并行转换和分段转换三种，如表 6-24 所示。

表 6-24 系统之间的转换方式对比

转换方式	描 述	备 注
直接转换	是指在确定新系统运行无误时，立刻启用新系统，终止旧系统运行	这种转换方式对人员、设备费用很节省。这种方式一般适用于处理过程不太复杂、数据不太重要的场合
并行转换	是指新旧系统并行工作一段时间，经过一段时间的考验以后，新系统正式替代旧系统。对于较复杂的大型系统，它提供了一个与旧系统运行结果进行比较的机会，可以对新旧两个系统的时间要求、出错次数和工作效率给以公正的评价	这是一种经常使用的转换方式。由于与旧系统并行工作，因此消除了尚未认识新系统之前的紧张和不安。其特点是安全、可靠，但费用和工作量都很大，因为在相当长的时间内系统要两套班子并行工作
分段转换	又称逐步转换、向导转换、试点过渡法等。这种转换方式实际上是直接转换与并行转换方式的结合。在新系统全部正式运行前，一部分一部分地代替旧系统。那些在转换过程中还没有正式运行的部分，可以在一个模拟环境中继续试运行	这种转换方式既保证了可靠性，又不至于费用太大。但是这种分段转换要求子系统之间有一定的独立性，对系统的设计和实现都有一定的要求，否则将无法实现这种分段转换的设想

(21) D。建议使用卡诺图进行化简：

$$\begin{aligned} F(A, B, C, D) &= \sum m(5) + \sum d(1, 7, 11, 13, 15) \\ &= \overline{A}\overline{B}\overline{C}D + \overline{A}B\overline{C}D + A\overline{B}\overline{C}D + ABC\overline{D} + ABCD = BD \end{aligned}$$

(22) C。施密特触发器是具有滞后特性的数字传输门，电路具有两个阈值电压，分别称为正向阈值电压和反向阈值电压。与双稳态触发器和单稳态触发器不同，施密特触发器属于“电平触发”型电路，不依赖于边沿陡峭的脉冲。施密特触发器常应用于对脉冲波形整形、阈值探测以及脉冲展宽等场合。

(23) C。对于本试题的答案可采用尝试法进行选择。通常在这类选择题中，按照答案在选项中被选中的概率，建议读者按“C、D、B、A”的顺序进行计算，以尽可能节约宝贵的考试时间。

根据“每一种数制都使用位置表示法，即处于不同位置的数符所代表的值不同，与它所在位置的权值有关”，在六进制中，将表达式 $(251)_6 - (53)_6$ 转换成十进制数进行计算：

$$\begin{aligned}(251)_6 - (53)_6 &= (2 \times 6^2 + 5 \times 6^1 + 1 \times 6^0) - (5 \times 6^1 + 3 \times 6^0) = 103 - 33 = 70 \\ &= 1 \times 6^2 + 5 \times 6^1 + 4 \times 6^0 = (154)_6\end{aligned}$$

$$\begin{aligned}\text{在七进制中, } (251)_7 - (53)_7 &= (2 \times 7^2 + 5 \times 7^1 + 1 \times 7^0) - (5 \times 7^1 + 3 \times 7^0) \\ &= 124 - 38 = 96 \\ &= 1 \times 7^2 + 6 \times 7^1 + 5 \times 7^0 = (165)_7\end{aligned}$$

$$\begin{aligned}\text{在八进制中, } (251)_8 - (53)_8 &= (2 \times 8^2 + 5 \times 8^1 + 1 \times 8^0) - (5 \times 8^1 + 3 \times 8^0) \\ &= 169 - 43 = 126 \\ &= 1 \times 8^2 + 7 \times 8^1 + 6 \times 8^0 = (176)_8\end{aligned}$$

$$\begin{aligned}\text{在十六进制中, } (251)_{16} - (53)_{16} &= (2 \times 16^2 + 5 \times 16^1 + 1 \times 16^0) - (5 \times 16^1 + 3 \times 16^0) \\ &= 593 - 83 = 510 \\ &= 1 \times 16^2 + 15 \times 16^1 + 4 \times 16^0 = (1F4)_{16}\end{aligned}$$

由以上分析可知, 本题在八进制时表达式 $251 - 53 = 176$ 成立, 因此应选择选项 C。

(24) D。设计只由数字电路组成的 PCB 的地线系统时, 将接地线做成闭环路可以明显提高抗噪声能力。

在低频电路中, 通常信号的工作频率小于 1 MHz, 它的布线和元件间的电感影响较小, 而接地电路形成的环流对于干扰影响较大, 因而应采用一点接地。

当信号工作频率大于 10 MHz 时, 地线阻抗变得很大, 此时应尽量降低地线阻抗, 应采用就近多点接地。

当工作频率在 1~10 MHz 时, 如果采用一点接地, 则其地线长度不应超过波长的 1/20, 否则应采用多点接地法。

电路板上既有高速逻辑电路, 又有线性电路时, 应使它们尽量分开, 而两者的地线不要相混, 分别与电源端地线相连。要尽量加大线性电路的接地面积。

若接地线很细, 接地电位则随电流的变化而变化, 致使电子设备的定时信号电平不稳, 抗噪声性能变坏。因此应将接地线尽量加粗, 使它能通过三倍于 PCB 的允许电流。如有可能, 接地线的宽度应大于 3 mm。

(25) C。IP 核是指具有知识产权的、功能具体、接口规范且可在多个集成电路设计中重复使用的功能模块, 是实现系统芯片(SOC)的基本构件。

IP 核模块有行为(Behavior)、结构(Structure)和物理(Physical)三级不同程度的设计。对应描述功能行为的不同, IP 软核(Soft IP Core)完成行为描述, IP 硬核(Hard IP Core)基于物理描述并经过工艺验证, IP 固核(Firm IP Core)完成结构描述。

(26) A。选项 A 的说法是错误的, 因为在一片嵌入式处理器基础上添加电源电路、时钟电路和存储器电路, 就构成了一个嵌入式核心控制模块, 其中操作系统和应用程序都可以固化在 ROM 中。

(27) D。板级支持包(BSP, 也称为硬件抽象层 HAL)一般包含相关底层硬件的初始化、数据的输入/输出操作和硬件设备的配置等功能, 它主要具有以下两个特点:

① 硬件相关性。因为嵌入式实时系统的硬件环境具有应用相关性, 而作为上层软件与硬件平台之间的接口, BSP 需为操作系统提供操作和控制具体硬件的方法。

② 操作系统相关性。不同的操作系统具有各自的软件层次结构, 因此不同操作系统具有特定的硬件接口形式。

(28) A。实时系统都要求确定的响应时间,但是对于不同的系统,响应时间的要求也不同。即使是同一个系统,在处理不同性质的任务时,对于响应时间也会有不同的要求,它取决于系统运行结果对于时间的依赖程度。

关于嵌入式系统实时调度技术分类中,抢占式调度技术和非抢占式调度技术的比较见表 6-25。

表 6-25 嵌入式系统实时调度技术对比

调度方式	定 义	优 点	缺 点
抢占式调度	每个任务都有优先级,任何时候都具有最高优先级且已启动的任务先执行	实时性好,调度算法较简单,可保证高优先级任务的时间约束	上下文切换多
非抢占式调度	不允许任务在执行期间被中断,任务一旦占用嵌入式微处理器就必须执行完毕或自愿放弃	上下文切换少	微处理器有效资源利用率低,可调度性差

静态表驱动策略是一种离线调度策略,是指在系统运行前根据各任务的时间约束及关联关系,采用某种搜索策略生成一张运行时刻表,这张时刻表一旦生成后就不再发生变化。

静态优先级调度(也称为固定优先级调度)是指任务的优先级定义后,在任务运行过程中优先级不会发生变化。

动态优先级调度是指任务的优先级可以随着时间或系统状态的变化而发生变化。

(29) C; (30) D。CRC-16 标准规定的生成多项式为 $G(X) = X^{16} + X^{15} + X^2 + 1$, 它产生的校验码是 16 位。如果 CRC 的生成多项式为 $G(X) = X^4 + X + 1$ (对应的二进制数是 10011), 信息码字为 10110, 则 CRC 校验码的计算过程如下:

$$\begin{array}{r}
 10110\ 0000 \\
 10011 \\
 \hline
 101\ 00 \\
 100\ 11 \\
 \hline
 1\ 1100 \\
 1\ 0011 \\
 \hline
 1111
 \end{array}$$

因此,计算出的 CRC 校验码为 1111。

(31) B。如果信息位长度为 K , 监督码长度为 r , 则要指示 1 位错的 $N = K + r$ 个可能位置, 即纠正 1 位错, 则必须满足如下关系: $2^r - 1 \geq K + r$ 。

本题中, $K = 5$, 则 $2^r \geq 5 + r + 1 = 6 + r$, 当 $r = 4$ 时, 不等式成立且满足题目中“最少校验位”的要求。

(32) B。MCS-51 的寻址方式较多, 表 6-26 归纳了在寻址方面考试中可能出现的一些考核知识点, 考生应熟记之。

表 6-26 MCS-51 常见的几种寻址方式

序号	寻址方式	指令中操作数	例 子
1	直接寻址	直接以单元地址形式出现	MOV A, 63H
2	寄存器寻址	存放在寄存器中	MOV A, R0
3	寄存器间接寻址	寄存器中存放操作数的地址	MOV A, @R0 或 MOVX A, @DPTR
4	立即数寻址	直接给出	MOVB A, #5EH

(33) D。在 200 MHz 外频条件下，DDR 内存带宽可达 $200 \times 10^6 (64 \text{ bit}/8) \times 2 = 3.2 \text{ Gb/s}$ 。如果在 133 MHz 外频条件下，DDR 内存带宽达 $133 \times 10^6 (64 \text{ bit}/8) \times 2 = 2.1 \text{ Gb/s}$ 。

(34) A。并联比较型 A/D 转换器的最大优点是转换速率快。一个输出为 8 位的并联比较型 A/D 转换器的转换时间可以达到 50 ns 以下，是其他类型 A/D 转换器都无法做到的。

(35) C。建立时间是指将一个数字量转换为稳定模拟信号所需的时间。D/A 转换器通常使用建立时间来描述其速度，而不是用 A/D 转换中常用的转换速率。

(36) C；(37) A；(38) A。在实模式存储管理方案中：

- ① 全局变量 gvCh 和 gvShort 由于没有设置初始值，存放在 .bss 段中；
- ② 全局变量 gvInt 和 gvLong 设置了解情况初始值，存放在 .data 段中；
- ③ main 函数编译后的机器代码存放在 .text 段中；
- ④ 指针 p 和数组 array 都是 main 函数的局部变量，所以存放在栈空间中。
- ⑤ malloc 函数分配的存储空间，则位于堆空间当中。

(39) D；(40) B。在实现文件系统时，为加快文件目录的检索速度，可利用“文件控制块分解法”。其工作原理是查找文件内部号，而不是把文件控制块的所有内容都读入，这样在查找过程中所需读入的存储块减少，从而减少了访问磁盘的次数。

假设目录文件存放在磁盘上，每个盘块占 512 字节。通常将文件控制块分解成两部分，第 1 部分文件名占 8 字节，第 2 部分包括文件内部号和文件其他描述信息占 56 字节。如果某一目录文件共有 254 个文件控制块，那么采用分解法前，查找该目录文件的某一个文件控制块的平均访问磁盘次数为

$$64 \times \frac{254/2}{512} = 16 \text{ 次}$$

采用分解法后，查找该目录文件的某一个文件控制块的平均访问磁盘次数为

$$10 \times \frac{254/2}{512} + 1 = 4 \text{ 次}$$

采用这种方法访问文件，当找到匹配的文件控制块后，还需要进行一次磁盘访问，才能读出全部的文件控制块信息。若目录文件分解前占用 n 个盘块，分解后改用 m 个盘块存放文件名和文件内部号部分，那么要减少访问磁盘次数需满足的条件是： $m < n - 2$ 。

(41) D。RS-232-C 是由电子工业协会(EIA, Electronic Industries Association)制定的数据终端设备和数据电路端与设备连接的物理接口标准,属于国际标准化组织 ISO 的开放系统互连(OSI)模型中的最低层,即物理层的协议标准。它规定了接口的机械、电气、规程和功能四个特性。

RS-232-C 规定的机械特性是 25 针的插头/座,减去一些未定义的针外,实际上只定义了 20 根针的功能。用来连接两个设备至少要连接 3 根线,即信号地、发送数据和接收数据线。在采用 RS-232-C 连接计算机与终端的场合就只使用这 3 根线。

RS-232-C 的设计数据速率是 20 000 b/s,连接设备间的距离也有规定。为了实现更高的数据速率和更远的距离连接,EIA 又制定了另一个 RS-422 标准。该标准的电气特性与 RS-232-C 不同,不用公共地,采用双线平衡传输的方式,在同样的数据速率条件下,可达到较远的传输距离。在数据速率为 20 000 b/s 时,连接设备间的距离可扩展到原有 RS-232-C 标准的约 80 倍。

(42) A。从编程的角度看,ARM 微处理器的工作状态一般有两种,并可在两种状态之间切换:

- ① ARM 状态,此时处理器执行 32 位的、字对齐的 ARM 指令;
- ② Thumb 状态,此时处理器执行 16 位的、半字对齐的 Thumb 指令。

(43) C。I²C 总线在数据传送过程中共有开始信号、结束信号和应答信号三种类型信号。

开始信号: SCL 为高电平,SDA 由高电平向低电平跳变时,开始传送数据;

结束信号: SCL 为低电平,SDA 由低电平向高电平跳变时,结束传送数据;

应答信号: 为特定的低电平脉冲。

(44) B。降低嵌入式系统功耗设计的措施有: ① 采用低功耗器件; ② 采用高集成度专用器件; ③ 动态调整微处理器的时钟频率和电压,例如采用动态电压调整技术(DVS 技术); ④ 利用“节电”工作方式; ⑤ 实行电源管理; ⑥ 设计外部中断唤醒电路等。

(45) C。ADSL G. lite 的最高速率标准中上行速率为 512 kb/s,一个 100 MB 文件所含的比特数 $D=100 \times 1024 \times 1024 \times 8$,根据公式 $S=D/t$ 得

$$t=D/S=100 \times 1024 \times 1024 \times 8 \text{ bit}/(512 \times 1000 \text{ b/s})=1638.4 \text{ s}$$

(46) A。ICMP 协议和 IP 协议均属于 TCP/IP 协议簇的网络层协议,但 ICMP 报文封装在 IP 包中传送。

(47) B。引导加载程序 BootLoader 是嵌入式系统加电后运行的第一段软件代码。

嵌入式系统通常没有像 BIOS 的固化程序,所以整个系统的加载启动任务完全由 BootLoader 程序来完成。

位于只读存储器 ROM 中的 BIOS 在完成硬件检测和资源分配后,将硬盘主引导记录(MBR)的引导程序读到系统的内存中,然后将控制权交给它。

GRUB 程序是一种 BootLoader 引导程序。

板级支持包(BSP)主要包含 BootLoader 程序和设备驱动程序两方面内容。

(48) C。在 Linux 操作系统中,Shell 变量分为参数变量和状态变量两类,见表 6-27。

表 6-27 Shell 变 量

Shell 参数变量		Shell 状态变量	
\$0	命令名,可用它获得调用该程序的名字	\$?	上一个命令的返回代码,如果命令执行成功则返回真值,否则返回假值
\$1, \$2, ...	表示 Shell 程序的位置参数	\$\$	表示当前命令的进程标识数
\$#	表示位置参数的个数,不包括命令名	\$_	表示 Shell 执行的最近后台进程标识数
\$*	表示所有位置参量,即相当于 \$1、\$2、\$3	\$@	与“\$*”基本相同,但当用双引号转义时,“\$@”还是能分解成多个参数,但“\$*”则合并成一个参数
\$-	表示 Shell 标志位组成的字符串,可以由 Shell 传递来,或由 set 命令设置		

注意到试题中“用户键入的命令参数的个数”关键信息,查表 6-27 可知,空缺处的正确答案是 \$#。

本试题中使用 case 语句实现编程中多选一的控制结构。cat 命令用于查看文件内容,从键盘读取数据,合并文件等。格式:cat [-b][-A][-E][-T][-n][-s][-v] 文件名。例如,“\$ cat -n file”命令将显示出 file 文件中的行号。

(49) A。源程序目录 uclinux-44b0ii 是 Linux 2.4.x 内核源代码文件目录,下面简要介绍目录文件的分布情况。

① kernel 目录下文件实现了大多数 Linux 系统的核心函数,其中最重要、最主要的文件当属 sched.c。

② mm/nommu 目录, Linux 中独立于 CPU 体系结构特征的内存管理文件几乎都集中在此目录下,如页式存储管理、内存的分配和释放等等。在 nommu 目录下是没有 MMU 的内存管理。

③ fs 目录源程序涵盖各种类型的文件系统和文件操作。

④ arch 目录包括与 CPU 类型相关的子目录和文件。

⑤ include 目录容纳 Linux 源程序的所有头文件(header file)。

⑥ net 目录存放的是和 Linux 网络相关的 C 文件。

⑦ drivers 目录存放各种设备的驱动程序。

(50) A。Linux 系统 smb.conf 文件中, max log size=0 语句用于设置日志文件的最大容量,大小以 KB 为单位。当 size=0 时,表示对日志文件的大小没有限制。

(51) C。在虚拟存储系统中常使用相联存储器进行管理,相联存储器的最主要特征就是按内容寻址。

(52) B。文件系统中,设立打开文件(Open)系统功能调用的基本操作是把文件的控制管理信息从辅存读到内存;设立关闭文件(Close)系统功能调用的基本操作是把文件当前的控制管理信息从内存写入磁盘。

(53) D。在一请求分页存储管理系统中,主存容量为 1 MB,由于主存容量被划为 512 块,因此可得到每块容量 $= 1 \text{ MB} / 512 = 1024 \text{ KB} / 512 = 2 \text{ KB}$ 。十进制逻辑地址 6186,可拆解成 $6186 = 2048 \times 3 + 42$,由此得该逻辑地址的页号为 3。查该作业的页表知,该页号的块状态为“1”,表示未被主存装入,由此将产生缺页中断。

(54) C。嵌入式系统支持的内存块由大到小分别是段(1 MB)、大页(64 KB)、小页(4 KB)和极小页(1 KB)。大页可分成大小为 16 KB 的子页,小页可分成大小为 1 KB 的子页。

(55) A。嵌入式系统对应用软件的开发和编程语言的选择提出了不同的要求,需要考虑如下一些因素:

① 实时性。许多嵌入式系统要求具有实时处理的能力,这种实时性主要是靠软件层来体现的。软件对外部事件做出反应的时间必须快,在某些情况下还要求是确定的、可重复实现的,不管系统当时的内部状态如何,都是可以预测的。

② 并发性。有些嵌入式系统要求支持多任务,能够处理并发事件。

③ 有限的软/硬件资源。在嵌入式系统当中,资源是很有限的,无论是处理器的运算速度、存储器的容量,还是 I/O 设备的种类,都比不上通用的计算机,甚至连最基本的电力供应,在嵌入式系统中也是一项约束条件,因为许多系统都是采用电池供电。这就使得嵌入式软件在时间和空间上都受到了严格的限制。

④ 涉及底层软件的开发。在无操作系统的情形下,嵌入式软件的开发是直接在硬件平台上进行的,需要直接对硬件进行控制;在有操作系统的情形下,需要先移植一个操作系统,并提供必要的 MI 函数,然后在操作系统的基础上开发应用程序。但不管是哪一种情形,都需要对底层的软件和硬件进行操作,包括引导加载程序的编写、设备驱动程序的编写、对设备控制器的操作等。

⑤ 需要交叉编译。嵌入式软件的开发环境与运行环境是不同的,需要交叉编译工具。

(56) C。嵌入式程序代码的生成过程是:高级语言代码→汇编语言代码→目标代码→二进制代码。

(57) C。栈帧是指在调用一个函数时,系统会自动地分配一块内存区域给这个函数,用来保存运行上下文、形参变量和局部变量等信息。

栈帧是在函数调用时分配的,当函数调用结束之后,相应的栈就会被释放。

(58) A。当一次函数调用发生时,其执行过程可归纳为以下五个步骤:

① 在内存的栈空间中为其分配一个栈帧,用来存放该函数的形参变量和局部变量。

② 把实参变量的值复制到相应的形参变量中。

③ 将控制流转移到该函数的起始位置。

④ 该函数开始执行。

⑤ 当这个函数执行完以后,控制流和返回值返回到函数调用点。

(59) D。外部变量的作用域是整个程序(全局变量),它由编译程序在编译时给其分配空间,属于静态分配变量,隐含类别是 extern(外部)存储类。引进外部变量的原因如下:一是程序运行外部变量的值是始终存在的;二是外部变量可以在所有函数间共享。

(60) B。如果嵌入式微处理器的 I/O 接口用来控制电机等噪声器件,则在 I/O 接口与噪声源之间应加隔离(增加 π 形滤波电路)。

(61) B。对于源程序的质量要求,最基本的就是正确性和可靠性,除此之外,更要注重程序的易使用性、易维护性和易移植性。易使用性就是要对用户友好,便于用户使用,做到少培训甚至零培训;易维护性包括易理解性、易测试性和易修改性;移植性则指程序从某一环境移植到另一环境的能力。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,定义每一个模块时尽可能少地显露其内部的处理。信息隐蔽原则对提高软件的可修改性、可测试性和可移植性都有重要的作用。

(62) D。软件可移植性是指与软件可从某一环境移植到另一环境的能力有关的属性。

通常高级语言具有较好的可移植性,因此编辑时应尽量用高级语言编写系统中对效率要求不高的程序部分。采用减少注释、选择时间/空间效率高的算法都不能提高软件的可移植性。

(63) A。在高级程序设计语言中,“语句”用于描述程序中的运算步骤、控制结构及数据传输。

程序设计语言的语法是语言的外观。给出语言的“语法”意味着给出语句、声明和其他语言结构的书写规则。“语义”则表示不同的语法结构的含义。在程序语言的手册中,语言的描述都是围绕着语法结构展开的。通常,先给出各种语句结构的语法,然后给出对应于该结构的语义以描述内在含义。

“语用”是指关于程序与使用者之间的关系。

(64) A。硬件是嵌入式系统运行的载体,也是嵌入式系统的基础。嵌入式系统硬件为嵌入式系统软件提供了执行环境,但也限定了嵌入式系统软件能够访问的资源。

嵌入式系统所能完成的功能首先可从硬件上得以体现。嵌入式系统的硬件设计是在嵌入式系统软/硬件划分的基础上,对划分为硬件部分的功能单元所进行的设计。

嵌入式系统硬件的选择包括硬件平台和嵌入式处理器的选择、外围设备的选择和接口电路的选择。其中,硬件平台在选择时应考虑:

- ① 如果已有的系统实现了相似的功能,重用该结构是个很好的选择。
- ② 如果这是个全新的项目,考虑这些功能是否能用一个处理器来实现。单个处理器是最容易实现和调试的。
- ③ 如果该应用需要用多个处理器,最好选用能够满足需求的最少处理器实现。
- ④ 在多处理器设计中,把控制和管理用一个处理器来实现,这样可简化操作。系统中其他处理器处理系统中的工作负载。

⑤ 按照这种方式设计的多处理器系统可以从一个小的系统扩充为一个大的系统,通过增加处理器来实现。因此客户可以从简单的处理器入手,随着系统负载的增加而增加。

(65) B。在嵌入式开发过程中,有主机和目标机的角色之分。主机是执行编译、链接和定址过程的计算机;目标机是指运行嵌入式软件的硬件平台。首先需把应用程序转换成可以在目标机上运行的二进制代码。这一过程包含三个步骤:编译、链接和定址。

① 编译过程由交叉编译器实现。所谓交叉编译器,就是运行在一个计算机平台上并为另一个平台产生代码的编译器。

② 编译过程产生的所有目标文件被链接成一个目标文件,称为链接过程。

③ 定址过程会把物理存储器地址指定给目标文件的每个相对偏移处。该过程生成的

文件就是可以在嵌入式平台上执行的二进制文件。

(66)~(70) D、B、C、A、D。

参考译文：在能够使用企业 EDM 之前，确定局域网上文档的位置可能很难，在广域网上几乎是不可能的。随着协同工作的流行以及更多的用户通过电子邮件和 Internet 共享文档，企业 EDM 已成为必备之物。

很多行业发现其最大的竞争优势是控制文档生命周期的能力。EDM 系统管理从文档的生成、修改、储存、检索到传输路径和 workflow 这一生命周期。它们也允许各用户协同地处理文档。

信息的重复使用能力是 EDM 的主要优点，同时也是生成驱动业务过程重建范例的关键。

(71)~(75) A、C、B、C、D。

参考译文：尽管大量的工业资源和精力集中于开发最快的微处理器和最灵活的操作系统，但越来越多的思想正在转向计算机接口的发展。在输入设备、语音处理和虚拟现实方面的进展可导致人机交互方式的根本改变。未来的技术战场将在用户和裸机之间加入多层，以使接口尽可能透明。虚拟现实代表着接口的下一个演化方向。

2. 下午试题要点解析

试题一

【问题 1】(2 分)

这是一道要求读者掌握 I²C 串行总线接口电路中各个部件作用的电路分析题。本题的解答思路如下：

① I²C 串行总线有两根信号线：一根是双向的数据线 SDA；另一根是时钟线 SCL。总线上受控器件的串行数据线 SDA 都接到总线的 SDA 线上，各受控器件的时钟线 SCL 接到总线的 SCL 上。

② 在图 6-17 所示的电路中，电阻 R36、R37 的阻值为 10 k Ω ，分别接在 AT24C64、DS1302、PCF8574 芯片的时钟线 SCL 和数据线 SDA 上。

③ 由于 I²C 总线接口均采用漏极开路 (Open-drain) 或集电极开路 (Open-collector) 的方式来完成线与 (Wired AND) 功能，因此需在数据线 SDA、时钟线 SCL 上接入上拉电阻。可见 R36、R37 分别是 I²C 串行总线上双向的数据线 SDA 与时钟线 SCL 的上拉电阻。

④ 另外，设计 I²C 总线接口电路时需要注意，总线的驱动能力以所连接器件的负载电容量不超过 400 pF 为宜。

【问题 2】(4 分)

这是一道要求读者阅读 I²C 总线开始条件与停止条件的时序图，并理解 I²C 总线数据有效性、数据字节格式的综合分析题。本题的解答思路如下：

① 每一个 I²C 总线器件都通过惟一的地址进行识别，根据其特性，可作为发送器或接收器工作。

② 由于连接到 I²C 总线上的器件各不相同 (如有 CMOS 器件、NMOS 器件、TTL 器件等)，逻辑 0 或逻辑 1 的电平会根据电源电压的高低发生变化，因此，每传输一个位就产生一个时钟脉冲。

③ 在图 6-18 所示的时序图中, I²C 总线的开始条件是指当 SCL 为高电平时, SDA 产生由高电平到低电平的跳变。停止条件是指当 SCL 为高电平时, SDA 产生由低电平到高电平的跳变。

④ 开始和停止条件总是由主器件产生。在开始条件产生后, 总线被认为处于忙状态, 在完成数据传输产生停止条件后, 总线被认为处于空闲状态。因此(1)空缺处应填入“主器件”, (2)空缺处应填入“忙状态”。

⑤ I²C 总线数据有效性是指在时钟信号的高电平期间, SDA 传输线上的电平必须稳定, 只有在 SCL 传输线上的时钟信号为低电平时, 数据线上的高低电平才允许发生变化。因此(3)空缺处应填入“低电平”。

⑥ I²C 总线数据字节的格式规定, 每一个写到 SDA 传输线上的数据字节必须为 8 位长度, 每一次传输的字节数没有限制, 每传输一个字节必须跟一个应答位 ACK(见图 6-18), 传输字节时最高位在前(MSB-first)。如果接收器因为执行其他功能(如中断服务)而不能接收剩余的数据字节时, 接收器就保持时钟线 SCL 为低电平强制发送器进入等待状态, 只有当接收器准备接收其他字节并释放 SCL 传输线时, 数据传输才会继续进行。可见, (4)空缺处应填入“应答位 ACK”。

【问题 3】(4 分)

这是一道要求读者掌握 I²C 总线寻址的综合分析题。本题的解答思路源于图 6-18 所给出的 I²C 总线开始条件与停止条件的时序图。

① 试题关键信息“该字节地址决定主控器选择哪一个从器件”中, “该字节地址”是指图 6-18 中开始条件之后的“Address”。因此可推理出(5)空缺处应填入“开始条件”。

② I²C 总线还可以通过“广播”寻址方式同时寻址所有的 IC 器件。当使用广播寻址时, 理论上所有的 IC 器件都应该返回应答信号, 但当某个 IC 器件不需要进行数据传输时, 将忽略广播寻址而不作任何应答。广播寻址的第二个字节决定其后的操作。可见(6)空缺处应填入“广播寻址”。

③ 如果某个 I²C 器件需要获取数据, 则将发出应答信号并作为一个从接收器。因此(7)、(8)空缺处应分别填入“应答”、“从接收器”。

【问题 4】(5 分)

这是一道要求读者掌握汇编语言程序分析的综合题。本题的解答思路源于图 6-18 所给出的 I²C 总线开始条件与停止条件的时序图。根据该时序图及题干中已定义各个子程序可知, (9)空缺处应选择“LCALL WRBYT”; (10)空缺处应选择“LCALL CACK”; (11)空缺处应选择“LCALL RDBYT”; (12)空缺处应选择“LCALL MNACK”; (13)空缺处应选择“LCALL MACK”。

以下给出完整的模拟 I²C 总线接收 N 个字节数据的程序段(RDNBYTE)及其各条语句的功能注释。

RDNBYTE: MOV	R3, NUMBYTE	;取所需传送的字节数, 即 N 值
LCALL	START	;发送启动位
MOV	A, SLA	;读取寻址字节类型
LCALL	WRBYT	;发送寻址字节(读)
LCALL	CACK	;检查应答位

	JB	F0, RDNBYTE	;用户标志位为1时,转移至 RDNBYTE 标号处,否则向下执行
RDN:	MOV	R1, #MRD	;取接收数据缓冲区的首地址
RDN1:	LCALL	RDBYT	;读入一个字节
	MOV	@R1, A	;将所读入的内容送至接收数据缓冲区
	DJNZ	R3, ACK	;N个字节读完否?未完转至 ACK 标号处
	LCALL	MNACK	;N个字节读完后,发送非应答位
	LCALL	STOP	;发送停止位
	RET		;子程序返回
ACK:	LCALL	MACK	;发送应答位
	INC	R1	;接收数据缓冲区的地址指针增1
	SJMP	RDN1	;无条件转至 RDN1 标号处,继续读下一个字节数据

试题二

【问题1】(5分)

这是一道要求读者掌握分层数据流图中数据流的平衡原则的综合分析题。本题的解答思路如下:

① 本题的图 6-19 并不是完整的顶层数据流图,解答时需通过题干的说明信息以及第 0 层数据流图来分析顶层图并解答问题。

② 题干中提及的关键信息“房主可以在安装该系统时配置安全监控设备(如传感器、显示器和报警器等)”,在顶层数据流图(图 6-19)中这 3 个名词并没有完整地出现,仅出现了“报警器”一词。在图 6-19 中“电视机”实际上起题干中关键信息“显示器”的作用。结合图 6-19 中“传感器状态”这一输出数据流可判断出(D)空缺处应填入“传感器”这一外部实体。

③ 由于子层数据流图是其父数据流图中某一部分内部的细节图(或加工图),因此子层数据流图的输入/输出数据流应该保持一致,即在上一级数据流图中有几条数据流,其子图也一定有同样的数据流,而且它们的输送方向是一致的。

④ 在住宅安全系统第 0 层数据流图(图 6-20)中,加工 5(信息及状态显示)的输出数据流为“TV 信号”,其中“TV”是日常生活中电视机的英文缩写。在图 6-19 顶层数据流图中有一外部实体——“录像机、电视机”,因此可推理出图 6-19 中(B)空缺处的数据流就是“TV 信号”。

⑤ 根据数据流的方向可知,在图 6-19 所示的顶层数据流图中(C)空缺处的数据流属于输出数据流。在第 0 层数据流图(图 6-20)中,共有“TV 信号”、“电话拨号”、“告警类型”和“显示信息”四个输出数据流。在图 6-19 顶层数据流图中现已存在“TV 信号”、“电话拨号”和“显示信息”三个输出数据流,因此可推理出图 6-19 中流向“报警器”外部实体的数据流是“告警类型”,即(C)空缺处填写的数据流应是“告警类型”。

⑥ 根据数据流的方向可知,在图 6-20 所示的第 0 层数据流图中(E)空缺处的数据流属于输入数据流,且与“传感器监控”这一加工处理相关。在图 6-19 顶层数据流图中,外部实体“传感器”流入“住宅安全系统”的输入数据流是“传感器状态”,由此可推理出图 6-19 中(E)空缺处的数据流就是“传感器状态”。

⑦ 同理,根据数据流的方向可知,在图 6-19 所示的顶层数据流图中(A)空缺处的数据流属于输入数据流。在第 0 层数据流图(图 6-20)中流入“住宅安全系统”加工的输入数据流共有四个,分别是“用户配置请求”、“开始/停止”、“用户密码”和“传感器状态”等。在图 6-19 所示的顶层数据流图中现已存在有“开始/停止”、“用户密码”和“传感器状态”三个输出数据流,因此可推理出图 6-19 中由“控制面板”外部实体流出的数据流是“用户配置请求”,即(A)空缺处填写的数据流应是“用户配置请求”。

⑧ 将以上分析结果归纳整理成图 6-23 所示的完整的住宅安全系统顶层数据流图。

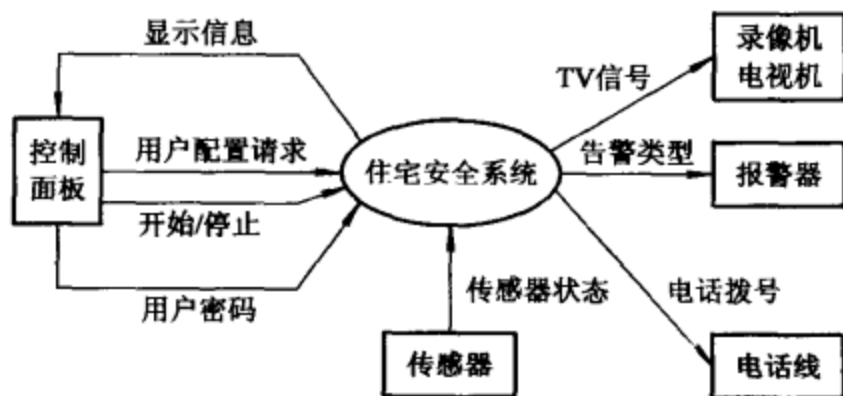


图 6-23 完整的住宅安全系统顶层数据流图

【问题 2】(3 分)

本试题的解答思路也是根据数据流图的数据流平衡原则进行分析的,详细的解答步骤如下:

① 由于子图是其父图中某一部分内部的细节图(加工图),因此子图的输入/输出数据流应该保持一致。其含义是:在上一级中有几条数据流,其子图也一定有同样的数据流,而且它们的输送方向是一致的。

② 在第 0 层数据流图(图 6-20)中,“4 传感器监控”加工模块有 1 条输入数据流——“传感器状态”和 3 条输出数据流——“电话拨号”、“传感器数据”、“告警类型”。

③ 在图 6-21 所示的加工 4 的细化图中,只有输出数据流“告警类型”,因此该加工图缺少了“传感器状态”、“电话拨号”及“传感器数据”这三条数据流。

④ 由于加工 4 的细化图(图 6-21)的内部结构、数据流向已清晰地给出,因此只需将“传感器状态”、“电话拨号”,“传感器数据”这 3 条数据流进行对号入座。

⑤ 根据常识可知,“电话拨号”是加工处理“4.5 拨号”的输出数据流;“传感器状态”是“4.4 读传感器”处理的输入数据流。

⑥ “传感器数据”应该是经“4.1 格式显示”处理过的数据流,所以作为“4.1 格式显示”的输出数据流。其中,“格式显示”处理是指将准备输出显示的传感器数据处理成第 0 层数据流图(图 6-20)中“5 信息与状态显示”能够接受的传感器数据格式。

⑦ 按照试题中“答题格式”的要求,将以上分析结果整理如下:

[4.5 拨号]→(电话拨号)

(传感器状态)→[4.4 读传感器]

[4.1 格式显示]→(传感器数据)

⑧ 另外,将以上分析结果归纳整理出完整的加工 4 的细化数据流图,如图 6-24 所示。

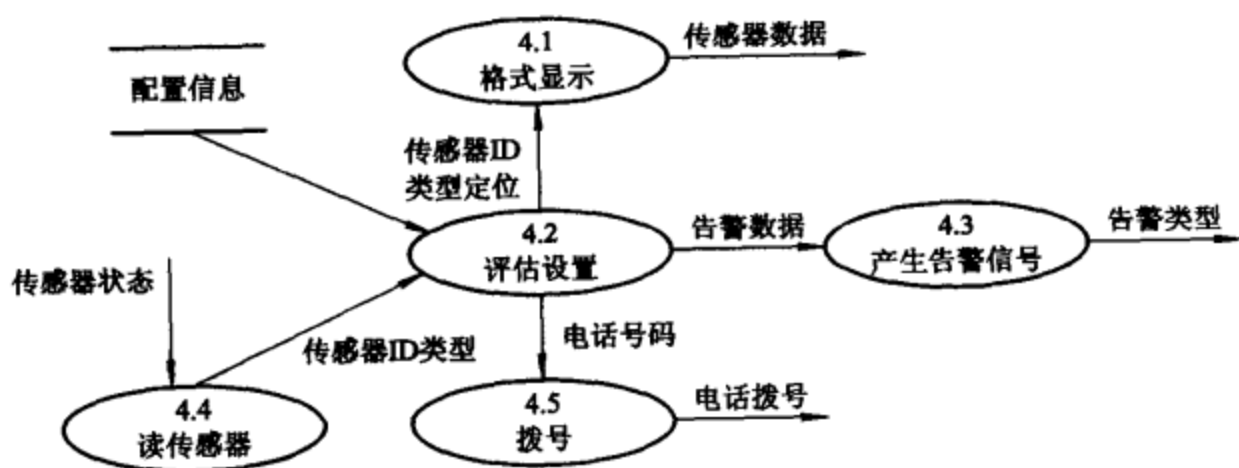


图 6-24 完整的加工 4 的细化数据流图

【问题 3】(3 分)

这是一道要求读者掌握数据流图(DFD图)中数据存储作用的综合分析题。本题的解答思路如下:

① 在数据流图(DFD图)中,数据存储是指逻辑上要求存储的数据,它不考虑具体数据的存储介质和技术手段。

② 在图 6-21 所示的加工 4 的细化数据流图中,可以阅读到“4.2 评估设置”这一加工处理用到了配置信息文件,由此可知,在第 0 层数据流图(图 6-20)中“4 传感器监控”加工将用到配置信息文件,即“4 传感器监控”加工处理与数据存储“配置信息”有关联。

③ 由题干关键信息“房主……可以在系统运行时修改配置”可知,此处的“修改配置”结果将存储到“配置信息”这一数据存储中。在第 0 层数据流图(图 6-20)中,“3 密码处理”是用于检验密码的,但是它只有一个输出数据流“检验 ID 信息”到“信息及状态显示”加工处理,并没有反馈回来的数据流,所以“检验 ID 信息”是已经验证通过的用户信息,用户输入密码应是在“3 密码处理”这个环节中进行验证的。如果密码验证是在“5 信息及状态显示”中进行的,那么从“5 信息及状态显示”应有一条不合法的用户数据流反馈到“密码处理”。可见,“密码处理”一定要用到数据存储“配置信息”中的用户名和密码。

④ 由以上分析可知,在第 0 层数据流图(图 6-20)中,由于输出到加工处理“5 信息及状态显示”的数据流是用来“检验 ID 信息”的,因此“5 信息及状态显示”加工处理也将与“配置信息”这一数据存储相关联。

【问题 4】(4 分)

这是一道要求读者掌握嵌入式实时系统分析任务的基本常识题。本题涉及到的知识点如下:

① 数据流图是系统分析阶段用于描述系统逻辑模型的图形描述工具,解决嵌入式系统“做什么”的问题。逻辑数据流图强调参与者所做的事情,可以帮助设计者决定需要哪些系统资源;为了运行系统,用户必须执行的活动;在系统安装之后,如何保护和控制这些系统。逻辑数据流图说明应该具有哪些加工和数据存储,而不关心这些加工和数据存储是如何实现的。

物理数据流图关注的是系统中的物理实体,以及一些具体的文档、报告和其他输入/输出硬拷贝。物理数据流图是用做系统构造和实现的技术性蓝图。物理数据流图需要说明加工和数据存储是如何实现的。

② 嵌入式实时系统分析阶段的主要任务是确定需要解决的问题或需要完成的目标及其约束,同时对实时系统的软/硬件做全面的分析,并对软/硬件做合理的分解,为实时系统的设计打下基础。实时系统的分析需要建模和仿真,以便系统分析人员估计“时间和大小”。(1)空缺处所填写的内容也可以从“建立系统模型时应明确体现……功能特点、约束条件等因素”中提取出“约束”以获得答案,(2)空缺处可填入“仿真”等内容。

③ 嵌入式实时系统环境中,系统建模就是用于描述所建立系统的软/硬件模型、优化系统描述的过程。

④ 建立系统模型时应明确体现性能描述、功能特点、技术指标、约束条件等因素。

其中,性能描述反映的是系统的整体面貌及体系结构,应明确地或隐含地说明系统 I/O 及相关的中间状态,以及相互之间的关系。

功能描述应明确表述系统各项功能特点与系统 I/O 及相关的中间状态之间的关系,以便对系统描述进行核实。

技术指标是评价系统质量的指标体系,常与价格、速度、字长和可靠性等具体项目有关。

约束条件应明确规定技术指标的适用范围、系统的工作环境要求及系统性能的缺陷、不足等。它们是确保系统正常工作的工作环境要求,是系统性能好坏的具体体现。

因此,(3)空缺处所填的内容是“性能描述”;(4)空缺处所填的内容是“技术指标”。

试题三

【问题 1】(10 分)

这是一道要求读者利用逻辑尺控制法进行循环程序设计的程序分析题。本题的解答思路如下:

① 在本案例的循环程序中,循环体又内嵌有多个不同的分支结构。每执行循环体一次,将根据事先规定好的标志位去执行具体的分支程序。试题要求采用逻辑尺控制法实现对于这种结构的循环程序设计。逻辑尺控制法首先应设计一把逻辑“尺”,即用字节、字、双字,甚至根据需要所设置的多字节中的各位表示不同的操作,如果需在循环中执行更多分支,也可采用多位组合。

② 题干中对执行程序段 Collect_task 和 Calculate_task 的顺序及次数做了如下约定:

Collect_task(2 次)→Calculate_task(1 次)→Collect_task(5 次)→Calculate_task(4 次)→Collect_task(2 次)→Calculate_task(2 次)

由于程序只简单执行两种不同的程序段,因此用一位“0”或“1”来分别标志转向执行两种不同的分支操作。执行程序段 Collect_task 和 Calculate_task 共有 16 次,因此采用一个字(16 位)来分别表示这两个程序段的执行顺序和次数。因此(1)空缺处可填入循环的总次数“16”,或者是等价的二进制数“00010000B”。

根据以上分析结果可设计出相应的逻辑尺,如图 6-25 所示。

15															0
0	0	1	0	0	0	0	0	1	1	1	1	0	0	1	1

图 6-25 逻辑尺定义

图 6-25 中, 阴影部分是程序段 Collect_task 执行的顺序和次数, 其余是程序段 Calculate_task 执行的顺序和次数。注意, 执行“Collect_task(2 次)”中第 1 次执行的 Collect_task 任务是放置在逻辑尺的最高位, 还是放置在逻辑尺的最低位, 由程序段中算术左移操作语句“SAL AX, 1”决定。将图 6-25 所示的逻辑尺的参数定义转换成等价的十六进制数, 0010 0000 1111 0011B=20F3H。因此(2)空缺处可填入“20F3H”, 或者是等价的二进制数“0010000011110011B”。

③ 程序中已给出代码(CODE)段定义伪指令“CODE SEGMENT”, 由于段定义伪指令的格式是:

〈段名〉SEGMENT [定位方式][连接方式][‘类别名’]

.....

〈段名〉ENDS

因此在模块结束伪指令“END START”之前需安排一条“CODE ENDS”, 即(8)空缺处需填入 CODE 段定义结束伪指令——“CODE ENDS”。

④ 由于程序段中已给出“功能号送入 AH 寄存器”及“中断返回”等关键信息, 由此可判断(7)空缺处需填入某一个中断类型码。由于软中断类型码通常取 20H~27H, 其中系统功能调用的通用格式是:

功能号送入 AH 寄存器

INT 21H

语句“MOV AH, 4CH”中“4CH”就是相应的功能号, 因此(7)空缺处需填入“21H”。

⑤ 结合逻辑尺控制算法的设计思想和试题中给出的指令及寄存器说明表, 仔细分析试题的程序段, 先填写(3)、(4)、(6)空缺处的标号内容, 最后完成(5)空缺处循环计数减 1 操作(DEC CL)的推理。以下给出该程序段的每条语句的详细解析。

N	EQU	14	; 定义循环的总次数
RULE	EQU	183CH	; 定义逻辑尺
CODE	SEGMENT		; CODE 段定义伪指令
	ASSUME	CS: CODE	; 段寄存器说明伪指令
START:	MOV	AX, RULE	; AX←逻辑尺标志
	MOV	CL, N	; CL←循环的总次数
LOP:	SAL	AX, 1	; 算术左移操作, CF←逻辑尺最高位
	JC	Calculate	; 当 CF=1 时, 转至 Calculate 标号处
Collect:	CALL	Collect_task	; 当 CF=0 时, 执行 Collect_task 程序段
	JMP	NEXT	; 无条件转至 NEXT 标号处
Calculate:	CALL	Calculate_task	; 执行 Calculate_task 程序段
NEXT:	DEC	CL	; 循环计数, CL←CL-1
	JNZ	LOP	; 若 CL≠0, 转至 LOP 标号处, 继续循环
	MOV	AH, 4CH	; 若 CL=0, 结束程序运行, 返回
	INT	21H	; 系统功能调用
CODE	ENDS		; CODE 段定义结束伪指令
	END	START	; 模块结束伪指令

【问题2】(3分)

这是一道要求读者掌握汇编程序基本功能及其执行过程的常识题。本题所涉及的知识如下：

① 汇编程序是一种系统软件，它的基本功能是将汇编语言源程序翻译成机器语言程序。

② 对于一个N行汇编语言代码的源程序，由于汇编指令中形成操作数地址的部分可能出现后面才会定义的符号，因此汇编程序一般至少需要两次扫描程序才能完成翻译过程。

③ 其中，第1次扫描的主要工作是通过固定的机器指令表MOT1定义符号的值并创建一个符号表ST，其中，ST记录了汇编时所遇到的符号的值。

④ 第2次扫描的任务是使用机器指令表MOT2产生目标程序。

【问题3】(2分)

这是一道要求读者掌握编译程序、解释程序基本功能的常识题。本题所涉及的知识如下：

① 编译程序的功能是把某些高级语言编写的源程序翻译成与之等价的目标语言程序。通常将编译程序的工作过程分为六个阶段，如表6-28所示。每个阶段的操作在逻辑上是紧密相连的，将源程序从一种表示形式一步步转换成目标代码。

表6-28 编译器六个工作阶段功能

名 称	主要任务	备 注
词法分析阶段	对源程序从前到后、从左到右逐个字符进行扫描，从中识别出一个个“单词”符号	这两个阶段的本质都是对源程序的结构进行分析
语法分析阶段	根据语言的语法规则将单词符号序列分解成各类语法单位，构造出相应的语法树	
语义分析阶段	对源程序进行类型分析，检查是否存在语义错误，并收集类型信息供代码生成阶段使用	这两个阶段的依据是语言的语义规则
中间代码生成阶段	根据语义分析的输出生成与具体机器无关的中间代码	
代码优化阶段	优化中间代码在时间和空间上的浪费问题	依据是程序的等价变换规则
目标代码生成阶段	把中间代码变换成特定机器上的绝对指令码、可重定位的指令代码或汇编指令代码	此阶段的工作与具体的机器相关
符号表管理	记录源程序中各个符号的必要信息，以辅助语义的正确性检查和代码生成	
出错处理	处理源程序中的静态错误和动态错误	

② 解释程序是另一种语言处理程序，在词法、语法和语义分析方面与编译程序的工作原理基本相同，但在运行用户程序时，它直接执行源程序或源程序的内部形式。

解释程序与编译程序的主要区别在于它不产生源程序的目标程序。

③ 解释程序通常可以分为两部分：第一部分是分析部分，包括通常的词法分析、语法分析和语义分析程序，经语义分析后把源程序翻译成中间代码，中间代码常采用逆波兰表示形式；第二部分是解释部分，用来对第一部分所生产的中间代码进行解释执行。

试题四

本试题考查考生实际的套接口网络编程能力。

(1) 本题对服务器主程序中(1)~(6)空缺处的解答思路如下：

① 管套接口(Socket)通过一组管套函数使应用程序在本地系统和远程系统之间建立通信信道来实现连接的管理和数据的传输。图 6-26 所示为虚电路服务中管套函数的调用顺序流程图，图 6-27 所示为数据报服务中管套函数的调用顺序流程图。

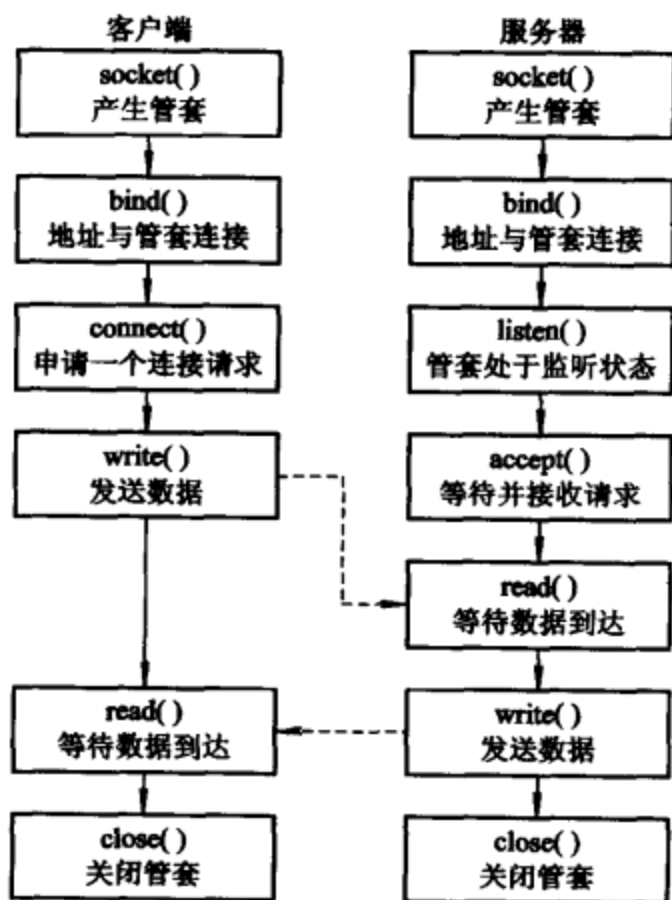


图 6-26 虚电路服务中管套函数的调用顺序流程图

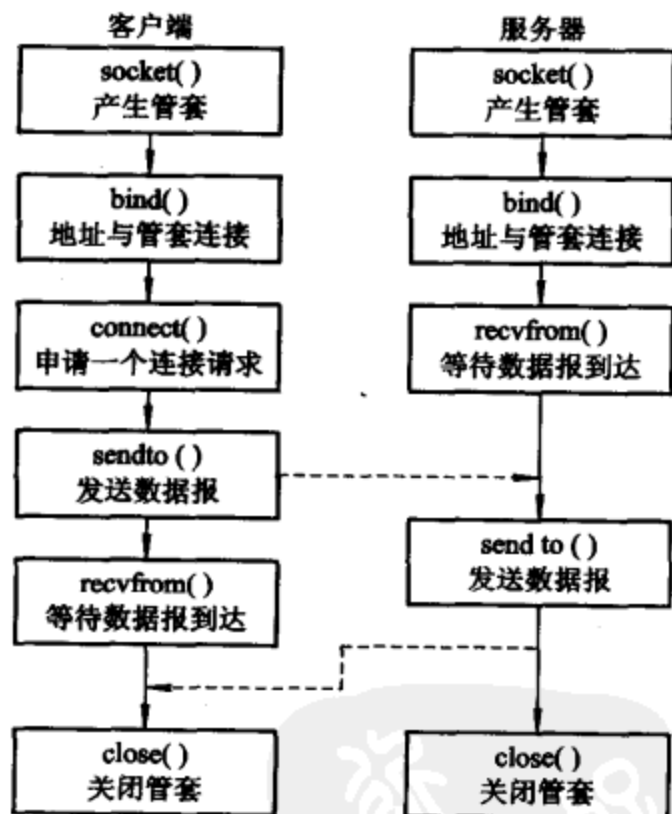


图 6-27 数据报服务中管套函数的调用顺序流程图

② 本试题中程序分为服务器主程序和服务器通信程序两部分。服务器主程序为 `int main(int argc, char * argv[])`。主程序流程如下：

定义监听套接字，连接套接字描述符→定义服务器监听地址→定义监听套接字并判误→地址结构初始化→设置合法的监听 IP 地址并转化为网络字节顺序格式→设置监听端口并转化为网络字节顺序格式→地址与套接字绑定并判误→监听套接字转化并判误→接收已

完成连接,创建新的连接套接字→连接失败否?是,则结束;否,则接收处理→关闭连接套接字→关闭监听套接字→结束。

③ 根据图 6-26 所示的虚电路服务中管套函数的调用顺序流程图可知,(1)空缺处调用 `socket(int domain, int type, int protocol)` 函数创建一个套接字描述符。其中, `domain` 为地址类型,本题中为 `AF_INET`; `type` 为套接字类型,本题中为 `SOCK_STREAM`; `protocol` 指定协议,本题中为 0。因此(1)空缺处需填入“`socket(AF_INET, SOCK_STREAM, 0)`”。

④ 根据(2)空缺处所在行的注释可知,此空缺处用于设置监听 IP 地址为任意合法地址,故属性为 `INADDR_ANY`。

⑤ (3)空缺处用于“设置监听的端口,并转化为网络字节顺序”,根据 `sockaddr_in` 类型的结构定义中“`unsigned short int sin_port`”可知,此处需调用的函数及属性是 `htons(SERVER_PORT)`。

⑥ 实现“将监听地址与用于监听的套接字绑定”功能需调用函数 `bind(int sockfd, struct sockaddr * myaddr, int addrlen)`。其中, `myaddr` 是本机地址,本题中可采用“`(struct sockaddr *)&servaddr`”获取; `addrlen` 为套接字地址结构的长度,本题中可通过函数“`sizeof(servaddr)`”获取。因此(4)空缺处填写的内容是“`listenfd, (struct sockaddr *)&servaddr, sizeof(servaddr)`”。

⑦ 实现“从监听套接字的连接队列中接收已经完成的连接,并创建新的连接套接字”功能需调用函数 `accept(int sockfd, struct sockaddr * addr, int * addrlen)`。其中, `sockfd` 为监听套接字,本题中为“`listenfd`”; `addr` 为客户机的地址; `addrlen` 为地址长度。本题在调用 `accept` 函数时,用常量 `NULL` 代替 `addr` 与 `addrlen` 表示无需取出客户机的地址信息。因此(5)空缺处填写的内容是“`accept(listenfd, NULL, NULL)`”。

⑧ 由“`close(listenfd);`”语句用于关闭监听套接字,以及监听套接字、连接套接字描述符的数据类型定义语句“`int listenfd, connfd`”可类比推理出,关闭连接套接字的语句是“`close(connfd);`”。

(2) 本题对服务器通信程序中(7)~(10)空缺处的解答思路如下:

① 服务器通信部分包括四个函数:

```
void serv_respon(int sockfd)
int read_requ(int sockfd, char * buf int size)
int get_char(int fd, char * ch)
int read_line(int fd, char * buf, int maxlen)
```

其中, `int read_requ(int sockfd, char * buf int size)` 程序的流程是:从套接字接收缓冲区中读出一行数据→从缓冲区 `buf` 中读出长度信息→从接收缓冲区中读出请求的内容部分。

② 由判断语句“`if (strncmp(inbuf, "", 6) == 0)`”可知, `read_requ` 程序每次从缓冲区 `buf` 中读取信息的长度为 6 个字节,因此(7)空缺处实现“从缓冲区 `buf` 中读出长度信息”的空缺内容是“`buf + 6`”。

③ 由程序中函数 `get_char(int fd, char * ch)` 的注释内容“`get_char` 的处理方式较为特殊,并不是每次调用 `read` 函数读一个字符,而是一次从缓冲区中读一块内容,再一次一个字符提交给函数 `read_line`,如果提交完了就再读一块,这样可以提高读缓冲区的效率”可

知, (8)空缺处循环条件设置与数据块长度 size 相关。再由语句“offset = 0”的注释“读出数据后, 将偏址置为 0”可知, (8)空缺处循环条件设置与偏址 offset 相关, 其填写的内容是“offset == size”。

④ 对于(9)空缺处的解答源于其注释行“将当前的字符取出, 并将偏址移向下一字符”, 注意是取出字符后才将偏址移向下一字符, 因此需填入的内容是“offset++”。如果填写的内容是“++ offset”, 则对“*ch = buf[++ offset];”语句的解释是, 先将偏址移向下一字符并将该字符取出。

⑤ 对于(10)空缺处的解答源于执行语句“break;”, 其需填入的内容是“ch == '\n'”, 用于判断结束条件。

试题五

【问题 1】(2 分)

这是一道要求读者掌握嵌入式系统软件编码步骤的简答题。本题所涉及的知识点如下:

在嵌入式系统设计过程中, 给定一份软件设计规格说明书后, 下一步的工作就是编写代码。通常编码工作可分为四个步骤:

- ① 给定源程序的标准格式, 制定编程规范。
- ② 准备编程环境, 包括软/硬件平台的选择、操作系统、编程语言、集成开发环境等。
- ③ 编写程序源代码。
- ④ 进行代码审查, 以提高编码质量。在审查时, 检查软件规格说明书与编码内容是否一致, 代码对硬件和操作系统资源的访问是否正确, 中断控制模块是否正确执行等。

【问题 2】(4 分)

这是一道要求读者掌握嵌入式系统软件平台选择的简答题。本题所涉及的知识点如下:

图 6-22 所示的(a)、(b)程序段的功能是完全一样的, 都是对一个结构体数组的各个元素进行初始化, 但采用两种不同的方法来实现。

图 6-22(a)所示的程序段采用的是数组下标的方法, 在定位第 i 个数组元素时, 需要将 i 乘以结构体元素的大小, 再加上数组的起始地址。

图 6-22(b)所示的程序段采用的是指针访问的方法, 先把指针 fp 初始化为数组的起始地址, 然后每访问完一个数组元素, 就把 fp 加 1, 指向下一个元素。

【问题 3】(4 分)

这是一道要求读者掌握嵌入式系统软件平台选择的简答题。本题所涉及的知识点如下:

在算术运算编码时, 需要仔细地选择和使用算术运算符。一般来说, 整数的算术运算最快, 其次是带有硬件支持的浮点运算, 而用软件来实现的浮点运算是比较慢的。因此在编码时需要遵守以下准则:

- ① 尽量使用整数(char、short、int 和 long)的加法和减法。
- ② 如果没有硬件支持, 尽量避免使用乘法。
- ③ 如果没有硬件支持, 尽量避免使用浮点数。
- ④ 编程时尽量避免使用除法。

【问题 4】(3 分)

这是一道要求读者掌握嵌入式系统软件平台选择的简答题。本题所涉及的知识点如下:

在进行嵌入式软件开发时,要注意对执行时间、存储空间和开发/维护时间这三种资源的使用进行优化。也就是说,代码的执行速度越快越好、系统占用的存储空间越小越好,软件开发和维护时间越少越好。具体来说,在编写代码时,需要做到以下六点编码准则:

- ① 保持函数短小,以提高代码的可靠性和可维护性。
- ② 封装代码,尽量消除代码之间的依赖性,提高对象的内聚性。
- ③ 消除冗余代码,避免出现类似一个变量初始化或设置一个变量后却从不使用的现象。
- ④ 减少实时代码,最好将对执行时间要求严格的代码转移到一个单独的任务或程序段中。
- ⑤ 编写优雅流畅的代码,例如使用 Tab 键为每个条件体或循环体进行有层次的缩进;将大括号放置在关键词下方的同列处;不要把小括号和关键词(if、while 等)紧贴在一起,要用空格隔开它们;不要把小括号和函数名紧贴在一起;等等。
- ⑥ 遵守代码编写标准并借助自动检查工具,以捕捉通过传统测试检查不到的各种问题。

另外,在嵌入式软件开发过程中,遵守编程规范,养成良好的编程习惯,将直接影响到所编写的代码的质量。编程规范主要涉及命名规则、编码格式及注释的书写这三方面的内容。

【问题 5】(2 分)

这是一道要求读者掌握嵌入式系统程序优化技术和程序执行时间的基本概念题。本题所涉及的知识点如下:

编译器对程序进行一定的优化是非常有必要的,高级语言的程序可以被转化为汇编语言的指令形式。优化技术是全面而非片面的。通常优化的方法有简化算术表达式、循环优化和废代码的清除等。

对嵌入式系统进行性能测试可以衡量系统的实时性,分析嵌入式软件的优劣,同时分析程序的执行时间,以帮助分析 CPU 功耗等特性。精确地测量程序的执行时间意义不大,可以通过以下三个概念来分析程序的执行时间。

- ① 平均执行时间:多次测量程序的执行时间取平均值。
- ② 最佳执行时间:大量输入序列的测试中记录嵌入式程序运行最短的那一个时间。
- ③ 最坏执行时间:程序的大量输入序列所产生的的最长的程序执行时间。有些情况下,产生最坏执行时间的输入数值并不容易确定,需要对程序的代码进行分析产生特殊的测试用例。

6.2.4 参考答案

1. 上午试题参考答案

表 6-29 给出了本模拟试卷问题(1)到问题(75)的参考答案,供读者练习时参考,以便查漏补缺。读者可按每空 1 分的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

2. 下午试题参考答案

表 6-30 给出了本模拟试卷试题一至试题五的参考答案,供读者练习时参考,以便查

漏补缺。读者也可依照所给出的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

表 6-29 参考答案及评分标准

问题号	参考答案	问题号	参考答案
(1)~(5)	C、D、D、C、D	(41)~(45)	D、A、C、B、C
(6)~(10)	B、A、B、B、D	(46)~(50)	A、B、C、A、A
(11)~(15)	B、A、D、A、D	(51)~(55)	C、B、D、C、A
(16)~(20)	C、A、B、A、D	(56)~(60)	C、C、A、D、B
(21)~(25)	D、C、C、D、C	(61)~(65)	B、D、A、A、B
(26)~(30)	A、D、A、C、D	(66)~(70)	D、B、C、A、D
(31)~(35)	B、B、D、A、C	(71)~(75)	A、C、B、C、D
(36)~(40)	C、A、A、D、B		

表 6-30 参考答案及评分标准

试题	问题与分值	参考答案及评分标准	自评分
一	【问题 1】 (2 分)	R36、R37 分别是 I ² C 串行总线上双向的数据线 SDA 与时钟线 SCL 的上拉电阻(2 分)	
	【问题 2】 (4 分)	(1) 主器件 (2) 忙状态 (3) 低电平 (4) 应答位 ACK(每空 1 分)	
	【问题 3】 (4 分)	(5) 开始条件 (6) 广播寻址 (7) 应答 (8) 从接收器(每空 1 分)	
	【问题 4】 (5 分)	(9) E 或 LCALL WRBYT (10) C 或 LCALL CACK (11) D 或 LCALL RDBYT (12) F 或 LCALL MNACK (13) B 或 LCALL MACK(每空 1 分)	
二	【问题 1】 (5 分)	(A) 用户配置请求 (B) TV 信号 (C) 告警类型 (D) 传感器 (E) 传感器状态(每空 1 分)	
	【问题 2】 (3 分)	[4.5 拨号]→(电话拨号)(1 分) (传感器状态)→[4.4 读传感器](1 分) [4.1 格式显示]→(传感器数据)(1 分)	
	【问题 3】 (3 分)	3 密码处理(1 分) 4 传感器监控(1 分) 5 信息及状态显示(1 分)	
	【问题 4】 (4 分)	(1) 约束 (2) 仿真 (3) 性能描述 (4) 技术指标(每空 1 分, (3)、(4)答案位置可互换)	

续表

试题	问题与分值	参考答案及评分标准	自评分
三	【问题 1】 (10 分)	(1) 16 或等价的二进制数“00010000B”(1 分) (2) 20F3H 或等价的二进制数“0010000011110011B”(2 分) (3) Calculate(1 分) (4) NEXT(1 分) (5) DEC CL(2 分) (6) LOP(1 分) (7) 21H(1 分) (8) CODE ENDS(1 分)	
	【问题 2】 (3 分)	(1) 机器语言程序(1 分) (2) 两次(2 分)	
	【问题 3】 (2 分)	编译程序的功能是把某些高级语言编写的源程序翻译成与之等价的目标语言程序(1 分) 解释程序与编译程序的主要区别是它不产生源程序的目标程序(1 分)	
四	【问题 1】 (15 分)	(1) socket(AF_INET, SOCK_STREAM, 0)(2 分) (2) INADDR_ANY(1 分) (3) htons(SERVER_PORT)(2 分) (4) listenfd, (struct sockaddr *)&servaddr, sizeof(servaddr)(2 分) (5) accept(listenfd, NULL, NULL)(2 分) (6) close(connfd)(1 分) (7) buf + 6(1 分) (8) offset == size(2 分) (9) offset++(1 分) (10) ch == '\n'(1 分)	
五	【问题 1】 (2 分)	① 给定源程序的标准格式, 制定编程规范 ② 准备编程环境, 包括软/硬件平台的选择、操作系统、编程语言、集成开发环境等 ③ 编写程序源代码 ④ 进行代码审查, 以提高编码质量(每小点 0.5 分)	
	【问题 2】 (4 分)	(a) 程序段采用的是数组下标的方法(1 分), 在定位第 i 个数组元素时, 需要将 i 乘以结构体元素的大小, 再加上数组的起始地址(1 分) (b) 程序段采用的是指针访问的方法(1 分), 先把指针 fp 初始化为数组的起始地址, 然后每访问完一个数组元素, 就把 fp 加 1, 指向下一个元素(1 分)	

续表

试题	问题与分值	参考答案及评分标准	自评分
五	【问题 3】 (4 分)	① 尽量使用整数(char、short、int 和 long)的加法和减法; ② 如果没有硬件支持, 尽量避免使用乘法; ③ 如果没有硬件支持, 尽量避免使用浮点数; ④ 编程时尽量避免使用除法(每小点 1 分)	
	【问题 4】 (3 分)	在编写代码时, 需要做到以下 6 点编码准则: ① 保持函数短小, 以提高代码的可靠性和可维护性; ② 封装代码, 尽量消除代码之间的依赖性, 提高对象的内聚性; ③ 消除冗余代码, 避免出现类似一个变量初始化或设置一个变量后却从不使用的现象; ④ 减少实时代码, 最好将对执行时间要求严格的代码转移到一个单独的任务或程序段中; ⑤ 编写优雅流畅的代码, 例如使用 Tab 键为每个条件体或循环体进行有层次地缩进; 将大括号放置在关键词下方的同列处等; ⑥ 遵守代码编写标准并借助自动检查工具, 以捕捉通过传统测试检查不到的各种问题(每小点 0.5 分)	
	【问题 5】 (2 分)	(1) 简化算术表达式(1 分) (2) 程序的大量输入序列所产生的的最长的程序执行时间(1 分)	

第7章 2006年下半年嵌入式系统设计师试题

7.1 上午试题

(考试时间 9:00—11:30 共 150 分钟)

请按下述要求正确填写答题卡。

(1) 在答题卡的指定位置上正确写入你的姓名和准考证号,并用正规 2B 铅笔在你写入的准考证号下填涂准考证号。

(2) 本试卷的试题中共有 75 个空格,需要全部解答,每个空格 1 分,满分 75 分。

(3) 每个空格对应一个序号,有 A、B、C、D 四个选项,请选择一个最恰当的选项作为答案,在答题卡相应序号下填涂该选项。

(4) 解答前请务必阅读例题和答题卡上的例题填涂样式及填涂注意事项。解答时用正规 2B 铅笔正确填涂选项,如需修改,请用橡皮擦干净,否则会导致不能正确评分。

例题

• 2007 年下半年全国计算机技术与软件专业技术资格(水平)考试日期是(88)月(89)日。

(88) A. 9 B. 10 C. 11 D. 12

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

因考试日期是“11 月 3 日”,故(88)选 C,(89)选 C,应在答题卡序号(88)下对 C 填涂,在序号(89)下对 C 填涂(参看答题卡)。

7.1.1 试题描述

• 若内存按字节编址,用存储容量为 $32K \times 8$ 比特的存储器芯片构成地址编号 0A0000H 至 0DFFFFH 的内存空间,则至少需要(1)片。

(1) A. 4 B. 6 C. 8 D. 10

• 某计算机系统由如图 7-1 所示的部件构成,假定每个部件的千小时可靠度 R 均为 0.9,则该系统的千小时可靠度约为(2)。

(2) A. 0.882 B. 0.951 C. 0.9 D. 0.99

• 设指令由取指、分析和执行三个子部件完成,每个子部件的工作周期均为 Δt ,采用常规标量单流水线处理机。若连续执行 10 条指令,则共需时间(3) Δt 。

(3) A. 8 B. 10 C. 12 D. 14

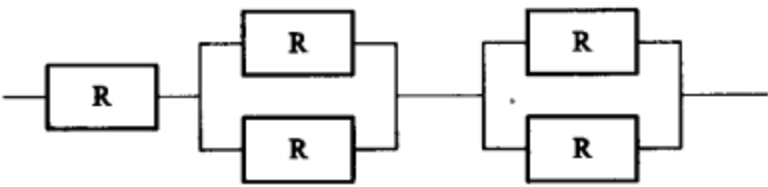


图 7-1 某计算机系统的组成结构图

• 某计算机的时钟频率为 400 MHz，测试该计算机的程序使用四种类型的指令。每种指令的数量及所需指令时钟数(CPI)如表 7-1 所示，则该计算机的指令平均时钟数为(4)；该计算机的运算速度约为(5)MIPS。

表 7-1 各种指令的数量及所需指令时钟数

指令类型	指令数目(条)	每条指令需时钟数
1	160 000	1
2	30 000	2
3	24 000	4
4	16 000	8

- (4) A. 1.85 B. 1.93 C. 2.36 D. 3.75
(5) A. 106.7 B. 169.5 C. 207.3 D. 216.2

• 某计算机指令字长为 16 位，指令有双操作数、单操作数和无操作数三种格式，每个操作数字段均用 6 位二进制表示，该指令系统共有 m 条($m < 16$)双操作数指令，并存在无操作数指令。若采用扩展操作码技术，那么最多还可设计出(6)条单操作数指令。

- (6) A. 2^6 B. $(2^4 - m) \times 2^6 - 1$
C. $(2^4 - m) \times 2^6$ D. $(2^4 - m) \times (2^6 - 1)$

• 以下不属于网络安全控制技术的是(7)。

- (7) A. 防火墙技术 B. 访问控制技术
C. 入侵检测技术 D. 差错控制技术

• “冲击波”病毒属于(8)类型的病毒，它利用 Windows 操作系统的(9)漏洞进行快速传播。

- (8) A. 蠕虫 B. 文件 C. 引导区 D. 邮件
(9) A. CGI 脚本 B. RPC C. DNS D. IMAP

• (10)确定了标准体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。

- (10) A. 标准化 B. 标准
C. 标准化法 D. 标准与标准化

• 某开发人员不顾企业有关保守商业秘密的要求，将其参与该企业开发设计的应用软件的核心程序设计技巧和算法通过论文向社会发表，那么该开发人员的行为(11)。

- (11) A. 属于开发人员权利不涉及企业权利 B. 侵犯了企业商业秘密权
C. 违反了企业的规章制度但不侵权 D. 未侵犯权利人软件著作权

• 计算机要对声音信号进行处理时,必须将它转换成为数字声音信号。最基本的声音信号数字化方法是取样-量化法。若量化后的每个声音样本用2个字节表示,则量化分辨率是(12)。

- (12) A. 1/2 B. 1/1024 C. 1/65 536 D. 1/131 072

• 某幅图像具有 640×480 个像素点,若每个像素具有8位的颜色深度,则可表示(13)种不同的颜色,经5:1压缩后,其图像数据需占用(14)(Byte)的存储空间。

- (13) A. 8 B. 256 C. 512 D. 1024

- (14) A. 61 440 B. 307 200 C. 384 000 D. 3 072 000

• 常见的软件开发模型有瀑布模型、演化模型、螺旋模型以及喷泉模型等。其中(15)模型适用于需求明确或很少变更的项目,(16)模型主要用来描述面向对象的软件开发过程。

- (15) A. 瀑布模型 B. 演化模型 C. 螺旋模型 D. 喷泉模型

- (16) A. 瀑布模型 B. 演化模型 C. 螺旋模型 D. 喷泉模型

• 软件能力成熟度模型(CMM)是目前国际上最流行、最实用的软件生产过程标准和软件企业成熟度的等级认证标准。该模型将软件能力成熟度自低到高依次划分为初始级、可重复级、已定义级、已管理级和优化级。从(17)开始,要求企业建立基本的项目管理过程的政策和管理规程,使项目管理工作有章可循。

- (17) A. 初始级 B. 可重复级 C. 已定义级 D. 已管理级

• 软件测试通常分为单元测试、组装测试、确认测试及系统测试四个阶段,(18)属于确认测试阶段的活动。

- (18) A. 设计评审 B. 代码审查 C. 结构测试 D. 可靠性测试

• 下面关于面向对象的描述正确的是(19)。

- (19) A. 针对接口编程,而不是针对实现编程
B. 针对实现编程,而不是针对接口编程
C. 接口与实现不可分割
D. 优先使用继承而非组合

• 在一个单CPU的计算机系统中,采用可剥夺式(也称抢占式)优先级的进程调度方案,且所有任务可以并行使用I/O设备。表7-2列出了三个任务T1、T2、T3的优先级和独立运行时占用CPU和I/O设备的时间。如果操作系统的开销忽略不计,则这三个任务从同时启动到全部结束的总时间为(20)ms,CPU的空闲时间共有(21)ms。

表7-2 各任务的优先级及独立运行时占用CPU和I/O设备的时间

任务	优先级	每个任务独立运行时所需的时间
T1	高	对每个任务: 占用CPU 10 ms, I/O 13 ms, 再占用CPU 5 ms
T2	中	
T3	低	

- (20) A. 28 B. 58 C. 61 D. 64

- (21) A. 3 B. 5 C. 8 D. 13

• 从表 7-3 关于操作系统存储管理方案 1、方案 2 和方案 3 的相关描述可以看出，它们分别对应(22)存储管理方案。

表 7-3 各操作系统存储管理方案说明

方案	说 明
1	在系统进行初始化的时候就已经将主存储空间划分成大小相等或不等的块，并且这些块的大小在此后是不可以改变的。系统将程序分配在连续的区域中
2	主存储空间和程序按固定大小单位进行分割，程序可以分配在不连续的区域中。该方案当一个作业的程序地址空间大于主存可以使用的空间时也可以执行
3	编程时必须划分程序模块和确定程序模块之间的调用关系，不存在调用关系的模块可以占用相同的主存区

- (22) A. 固定分区、请求分页和覆盖 B. 覆盖、请求分页和固定分区
C. 固定分区、覆盖和请求分页 D. 请求分页、覆盖和固定分区

• 若某逻辑门输入 A、B 和输出 F 的波形如图 7-2 所示，则 $F(A, B)$ 的表达式是(23)。

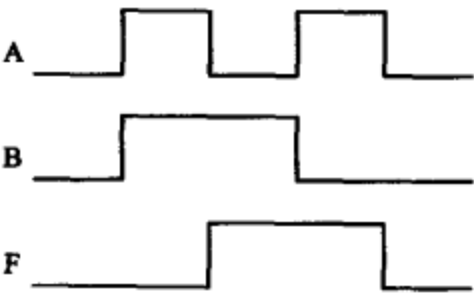


图 7-2 某逻辑电路波形图

- (23) A. $F = A \cdot B$ B. $F = A + B$ C. $F = A \oplus B$ D. $F = A \cdot \bar{B}$

• 一个 4 位的二进制计数器，由 0000 状态开始，经过 25 个时钟脉冲后，该计数器的状态为(24)。

- (24) A. 1100 B. 1000 C. 1001 D. 1010

• 稳压二极管构成的稳压电路的接法是(25)。

- (25) A. 稳压管与负载电阻串联
B. 稳压管与限流电阻并联
C. 限流电阻与稳压管串联后，与负载电阻串联
D. 限流电阻与稳压管串联后，与负载电阻并联

• 以下叙述中，不符合 RISC 指令系统特点的是(26)。

- (26) A. 指令长度固定，指令种类少
B. 寻址方式种类丰富，指令功能尽量增强
C. 设置大量通用寄存器，访问存储器指令简单
D. 选取使用频率较高的一些简单指令

• 通常所说的 32 位微处理器是指(27)。

- (27) A. 地址总线的宽度为 32 位 B. 处理的数据长度只能为 32 位
C. CPU 字长为 32 位 D. 通用寄存器数目为 32 个

• 在 32 位总线系统中,若时钟频率为 500 MHz,传送一个 32 位字需要 5 个时钟周期,则该总线系统的数据传送速率为(28) MB/s。

(28) A. 200 B. 400 C. 600 D. 800

• 在 CPU 和物理内存之间进行地址转换时,(29)将地址从虚拟(逻辑)地址空间映射到物理地址空间。

(29) A. TCB B. MMU C. CACHE D. DMA

• 评价一个计算机系统时,通常主要使用(30)来衡量系统的可靠性。

(30) A. 平均响应时间 B. 平均无故障时间(MTBF)
C. 平均修复时间 D. 数据处理速率

• 下列存取速度最快的是(31)。

(31) A. Flash 存储器 B. DRAM 存储器
C. 高速缓存(CACHE) D. 磁盘

• 除了 I/O 设备本身的性能外,影响嵌入式系统 I/O 数据传输速度的主要因素是(32)。

(32) A. Cache 存储器性能 B. 总线的传输速率
C. 主存储器的容量 D. CPU 的字长

• RS-232 标准在初期可以满足人们的要求,但当人们要求以更高的速率传送到更远的距离时,需要有新的标准。RS-449 就是为此目的而设计的,它的标准规格中,RS-422 标准是平衡式的,传输一个信号用两条线,逻辑信号用(33)表示,双线传输的主要优点是(34)。

(33) A. 正负 5 V 以上电平 B. 正负 3 V 以上电平
C. 两条线的电平差 D. 两条线的电平

(34) A. 冗余可靠 B. 抗干扰能力增强
C. 传输线少 D. 可并行传输信号

• 若曼彻斯特编码和差分曼彻斯特编码的波形图如图 7-3 所示,则实际传送的比特串为(35)。

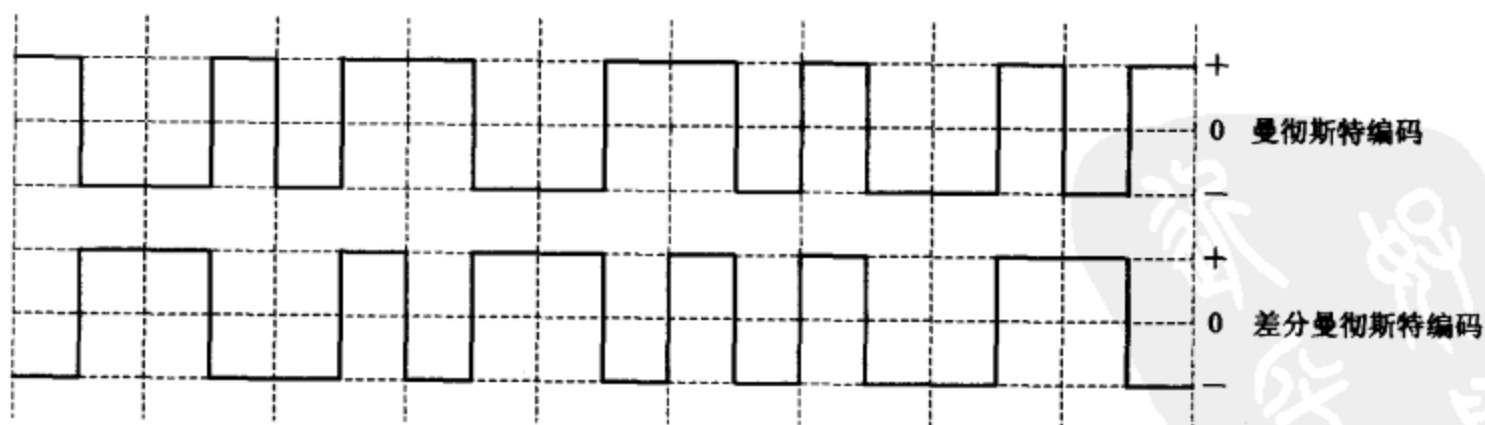


图 7-3 曼彻斯特编码和差分曼彻斯特编码波形图

(35) A. 011010110 B. 100101100
C. 100100100 D. 011010011

• 下面是关于 PCI 总线的叙述,其中(36)是错误的。

(36) A. PCI 总线支持 64 位总线

B. PCI 总线的地址总线与数据总线是分时复用的

C. PCI 总线是一种独立设计的总线, 它的性能不受 CPU 类型的影响

D. PC 机中不能同时使用 PCI 总线和 ISA 总线

• IEEE 80211. b 标准定义了使用跳频、扩频技术的无线局域网标准, 传输速率为 1 Mb/s, 2 Mb/s, 5.5 Mb/s 和(37)。

(37) A. 10 Mb/s

B. 11 Mb/s

C. 20 Mb/s

D. 54 Mb/s

• 电路板的设计主要分三个步骤, 不包括(38)这一步骤。

(38) A. 生成网络表

B. 设计印制电路板

C. 设计电路原理图

D. 自动布线

• 现代电子设计方法包含可测试设计, 其中(39)接口是 IC 芯片测试的标准接口。

(39) A. BIST

B. JTAG

C. UART

D. USB

• 多层印制电路板(4 层或者 4 层以上)比双面板更适合于高速 PCB 布线, 最主要的原因是(40)。

(40) A. 通过电源平面供电, 电压更稳定

B. 可以大大减小电路中信号回路的面积

C. 多层印制电路板工艺简单

D. 自动布线更容易

• 下面不符合数字电路(或者集成电路)的电磁兼容性设计方法的是(41)。

(41) A. IC 的电源及地的引脚较近, 有多个电源和地

B. 使用贴片元件, 不是用插座

C. IC 的输出级驱动能力应超过实际应用的要求

D. 对输入和按键采用电平检测(而非边沿检测)

• 嵌入式系统由硬件部分和软件部分构成, 以下(42)不属于嵌入式系统软件。

(42) A. 系统内核

B. 驱动程序

C. FPGA 编程软件

D. 嵌入式中间件

• 关于硬件抽象层, 以下描述中错误的是(43)。

(43) A. 硬件抽象层包括操作系统内核和驱动程序

B. 硬件抽象层将操作系统与硬件平台隔开

C. 硬件抽象层是一种软件

D. 硬件抽象层有利于系统的模块化设计

• 一个任务被唤醒, 意味着(44)。

(44) A. 该任务重新占有了 CPU

B. 它的优先级变为最大

C. 其任务移至等待队列队首

D. 任务变为就绪状态

• 关于实时操作系统(RTOS)的任务调度器, 以下描述中正确的是(45)。

(45) A. 任务之间的公平性是最重要的调度目标

B. 大多数 RTOS 调度算法都是可抢占式(可剥夺式)的

C. RTOS 调度器都采用了基于时间片轮转的调度算法

D. RTOS 调度算法只是一种静态优先级调度算法

- 在下列调度算法中, (46)算法不会出现任务“饥饿(starvation)”的情形。

(46) A. 时间片轮转算法 B. 先来先服务算法
C. 可抢占的短作业优先算法 D. 静态优先级算法

- 在实时操作系统中, 邮箱通信是一种(47)通信方式。

(47) A. 直接 B. 间接 C. 低级 D. 信号量

- 在某嵌入式操作系统中, 若 P、V 操作的信号量 S 的初值为 2, 当前值为 -1, 则表示等待信号量 S 的任务个数为(48)。

(48) A. 0 B. 1 C. 2 D. 3

- 在实时操作系统中, 两个任务并发执行, 一个任务要等待其合作伙伴发来消息, 或建立某个条件后再向前执行, 这种制约性合作关系被称为任务的(49)。

(49) A. 同步 B. 互斥 C. 调度 D. 执行

- 下面关于存储管理的叙述中, 正确的是(50)。

(50) A. 在嵌入式微处理器当中, 都配备有存储管理单元 MMU
B. 在嵌入式系统中, 内核空间和用户空间必须是两个相互独立的地址空间
C. 在有些嵌入式系统中, 甚至不提供存储管理功能
D. 在虚存系统中, 只要磁盘空间无限大, 任务就能拥有任意大的编址空间

- 页式存储管理当中的页面是由(51)所感知的。

(51) A. 用户 B. 操作系统 C. 编译系统 D. 链接程序

- 文件系统的主要功能是(52)。

(52) A. 实现对文件的按名存取 B. 实现虚拟存储
C. 提高外存的读写速度 D. 用于保存系统文档

- 在 FAT16 文件系统中, 若每个簇(cluster)的大小是 2 KB, 那么它所能表示的最大磁盘分区容量为(53)。

(53) A. 2 MB B. 32 MB C. 64 MB D. 128 MB

- 在下面的叙述中, (54)不是嵌入式图形用户接口(GUI)的主要特点。

(54) A. 运行时占用的系统资源少 B. 模块化结构, 便于移植和定制
C. 可靠性高 D. 美观华丽, 图形算法复杂

- 以下叙述中正确的是(55)。

(55) A. 宿主机与目标机之间只需要建立逻辑连接即可
B. 在嵌入式系统中, 调试器与被调试程序一般位于同一台机器上
C. 在嵌入式系统开发中, 通常采用的是交叉编译器
D. 宿主机与目标机之间的通信方式只有串口和并口两种

- 关于汇编语言, 以下叙述中正确的是(56)。

(56) A. 汇编语言程序可以直接在计算机上运行
B. 汇编语言通常会提供控制伪指令, 用于控制汇编程序的执行流程
C. 在汇编语言中, 不能定义符号常量
D. 伪指令语句所指示的操作是在程序运行时完成的

- 在 32 位处理器上, 假设栈顶指针寄存器的当前值为 0x00FFFE8, 那么在执行完指令“push eax”(eax 为 32 位寄存器)后, 栈指针的当前值应为(57)。

- (57) A. 0x00FFFFE4 B. 0x00FFFFE6
C. 0x00FFFFEA D. 0x00FFFFEC

• 在 C 语言中, 设有数组定义: `char array[] = "China"`, 则数组 `array` 所占用的空间为(58)。

- (58) A. 4 个字节 B. 5 个字节 C. 6 个字节 D. 7 个字节

• 执行 C 程序代码“`int a=1; int b=0; int c=0; int d=(++a)*(c=1);`”后, `a`、`b`、`c`、`d` 的值分别为(59)。

- (59) A. 2, 0, 1, 2 B. 1, 0, 1, 1
C. 2, 0, 1, 1 D. 2, 0, 0, 2

• 关于 C 语言的函数参数, 下列叙述中正确的是(60)。

- (60) A. 只有在发生函数调用时, 函数中的形参才被分配内存单元
B. 实参对形参的数据传递是单向的“值传递”, 因此, 在被调用的函数当中, 没有办法去修改主调函数中任何变量的值
C. 形参变量所占用的内存单元位于堆中
D. 如果形参是数组名, 则传递的是每个数组元素的值

• (61)的做法不利于嵌入式应用程序的移植。

- (61) A. 在软件设计上, 采用层次化设计和模块化设计
B. 在软件体系结构上, 在操作系统和应用软件之间引入一个虚拟机层, 把一些通用的、共性的操作系统的 API 接口函数封装起来
C. 将不可移植的部分局域化, 集中在某几个特定的文件之中
D. 在数据类型上, 尽量直接使用 C 语言的数据类型

• 在大型实时系统开发项目中, 体系结构设计产生的部件列表包括下列部件:

- ① 输入信号预处理 ② 主控制过程 ③ 网络接口

非功能需求的可靠性应该在(62)部件设计中考虑。

- (62) A. ① B. ①和② C. ②和③ D. ①、②和③

• (63)是软件系统测试的典型输入。

- ① 合同书 ② 需求规格说明书 ③ 程序代码 ④ 系统设计说明书

- (63) A. ①和② B. ①、②和③
C. ①和③ D. ①、②、③和④

• 系统分析模型应明确体现的因素是(64)。

- ① 性能描述 ② 功能描述 ③ 约束条件 ④ 系统结构 ⑤ 动态模型

- (64) A. ①、②、③、④和⑤ B. ①、②、③和⑤
C. ①、②和④ D. ①、②、④和⑤

• 嵌入式系统软/硬件协同设计从目标系统构思开始, 经过(65)阶段完成。

- ① 需求描述 ② 软/硬件划分 ③ 硬件综合、接口综合、软件编译
④ 软/硬件集成 ⑤ 软/硬件协同仿真与验证

- (65) A. ①、②、③、④和⑤ B. ②、③、④和⑤
C. ②、④和⑤ D. ①、②、④和⑤

• Embedded system is (66) special computer system which is scalable on both

software and (67). It can satisfy the strict requirement of functionality, (68), cost, volume, and power consumption of the particular application. With rapid development of (69) design and manufacture, CPUs became cheap. Lots of (70) electronics have embedded CPU and thus became embedded systems. For example, PDAs, cellphones, point-of-sale devices, VCRs, industrial robot control, or even your toasters can be embedded system.

- | | |
|------------------------|-------------------------|
| (66) A. cost-oriented | B. application-oriented |
| C. efficiency-oriented | D. consumption-oriented |
| (67) A. program | B. application |
| C. kernel | D. hardware |
| (68) A. variability | B. security |
| C. reliability | D. responsibility |
| (69) A. IC | B. manual |
| C. flexible | D. emulational |
| (70) A. industrial | B. military |
| C. consumer | D. special |

• An embedded operating system is the (71) program that manages all the other programs in an embedded device after initial load of programs by a(an)(72). It normally guarantees a certain capability within a specified (73) size and (74) constraint as well as with application programs. It also normally has small foot print including initial boot loader, OS kernel, required device drivers, (75)for the user data and so forth. It has very-like-ly structure of a normal operating system however mainly differentiated by some factors such as type of pre-installed device, functional limits, taking designed job only.

- | | |
|------------------|----------------|
| (71) A. hardware | B. business |
| C. software | D. external |
| (72) A. driver | B. application |
| C. kernel | D. boot loader |
| (73) A. data | B. storage |
| C. bus | D. CPU |
| (74) A. time | B. format |
| C. controller | D. packet |
| (75) A. JAFFS | B. format |
| C. file systems | D. protocol |

7.1.2 参考答案

表7-4给出本份真题问题(1)至问题(75)的参考答案,供读者练习时参考,以便查缺补漏。读者可按每空1分的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

表 7-4 参考答案及评分标准表

问题号	参考答案	问题号	参考答案
(1)~(5)	C、A、C、B、C	(41)~(45)	C、C、A、D、B
(6)~(10)	B、D、A、B、C	(46)~(50)	A、B、B、A、C
(11)~(15)	B、C、B、A、A	(51)~(55)	B、A、D、D、C
(16)~(20)	D、B、D、A、B	(56)~(60)	B、A、C、A、A
(21)~(25)	D、A、C、C、D	(61)~(65)	D、D、A、B、A
(26)~(30)	B、C、B、B、B	(66)~(70)	B、D、C、A、C
(31)~(35)	C、B、C、B、D	(71)~(75)	C、D、B、A、C
(36)~(40)	D、B、D、B、B		

7.1.3 要点解析

(1) C。本试题类似于 2004 年下半年上午试卷公共试题(中级)的第 2 题。

解答本试题时,可先计算按字节编址的内存从 0A0000H 至 0DFFFFH 的地址空间所占的字节数。首先考虑该连续的地址空间应包括 0A0000H 该字节的地址,因此该地址空间是: $(0DFFFFH+1)-0A0000H=0E0000H-0A0000H=40000H$ 。

由于内存按字节编址,也就是一个字节占一个地址空间,因此 $40000H=2^{18}B=2^8KB=256KB$ 。若要用存储容量为 $32K \times 8\text{ bit}$ 的存储器芯片构成该 256 KB 的内存,设至少需要存储器芯片的片数为 N。

$$N=(256 \times 1024 \times 8)/(32 \times 1024 \times 8)=8$$

因此试题(1)的答案是选项 C。

(2) A。该计算机系统的可靠性结构是一个典型的串/并联结构,而且构成系统的 5 个部件的千小时可靠度均为 $R_1=0.9$,则系统整体的千小时可靠度 R 为

$$R=R_1 \times (1-(1-R_1)(1-R_1))^2=0.9 \times (1-(1-0.9)^2)^2=0.882。$$

(3) C。流水线技术其实质是一种任务的分解技术。把一件任务分解为若干按顺序执行的子任务,不同的子任务由不同的执行机构负责执行,而这些机构可以同时并行工作。在任一时刻,任一任务只占用其中一个执行机构,这样就可以实现多个任务的重叠执行,以提高工作效率。若指令流水线把一条指令分为取指、分析和执行三部分,且三部分的工作周期均为 Δt ,则该流水线的操作周期为 Δt 。

由于在此指令流水线上执行第 1 条指令是流水线的建立过程,是按顺序执行的,因此从开始执行第 1 条指令到获得其执行结果所需时间为 3 个基本操作时间之和,即 $(1+1+1)\Delta t=3\Delta t$ 。

若连续执行 10 条指令,则其余 9 条指令均可在该流水线的操作周期时间 Δt 内完成。因此,这 10 条指令全部执行完毕所需时间为 $3\Delta t+9 \times \Delta t=12\Delta t$,即本题的答案是选项 C。

(4) B; (5) C。计算机执行指令的平均时钟数为总时钟数除以总条数,即本试题所给出的表 7-1 中指令的算术平均值。

$$\begin{aligned}\text{计算机执行指令的平均时钟数} &= \frac{160\,000 \times 1 + 30\,000 \times 2 + 24\,000 \times 4 + 16\,000 \times 8}{160\,000 + 30\,000 + 24\,000 + 16\,000} \\ &= 1.9304\end{aligned}$$

故指令的平均时钟数为 1.93 个时钟周期。

该计算机的速度为平均每秒钟所执行的指令条数，即其时钟在 1 s 内有多少个指令的平均时钟数。MIPS 表示每秒百万条指令。

$$\text{计算机的运算速度} \approx \frac{400\text{ MHz}}{1.93} = 207.254\text{ MIPS}$$

(6) B。通常一条指令包括操作码和操作数两部分。其中，操作码用于说明指令的功能及操作性质，操作数用于说明操作码实施操作的对象。本试题中指令字长为 16 位，也就是说，一条指令最长是 16 位。对于双操作数指令来说，两个长度为 6 位的操作数字段共占去 12 位，剩余 $16 - 12 = 4$ 位可用做操作码，可设计出 $2^4 = 16$ 条这样的双操作数指令。同理可知，对于单操作数指令来说，它的操作码长度为 $16 - 6 = 10$ 位。

试题中已给出该指令系统共有 m 条 ($m < 16$) 双操作数指令，那么剩余的 $2^4 - m$ 位可以用来设计单操作数和无操作数的操作码。若采用扩展操作码技术，那么单操作数指令在原双操作数指令的 4 位操作码上可扩展 $10 - 4 = 6$ 位。因此，最多可设计出 $(2^4 - m) \times 2^6 - 1$ 条单操作数指令，计算式子中，“-1”是考虑到试题给定的条件——“并存在无操作数指令”，因此至少留一位用来扩展成无操作数指令。

(7) D。防火墙技术是建立在内外网边界上的过滤封锁机制，能够防止外网未经授权访问内网，能够防止外网对内网的攻击，也能防止内网未经授权访问外网。但是，仅使用防火墙不能有效地防止内网的攻击。

入侵检测系统能够实时应对来自内网已知的攻击，对未知攻击的检测能力较弱，且存在误报率高的缺点。如果能将入侵检测系统和防火墙等其他安全系统进行联动，就能够更加有效地防止网络攻击。

访问控制技术用于解决网络中身份认证、机密性、数据完整性和抗抵赖性等问题。例如，采用身份认证技术可解决通信或数据访问中对对方身份的认可，便于访问控制，授权管理。

差错控制技术是通过信息量冗余来提高数据通信可靠性的一种技术，它不属于网络安全控制技术。

(8) A；(9) B。冲击波 (Worm. Blaster) 病毒利用 Windows 操作系统的 RPC DCOM 漏洞进行快速传播，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、无法正常上网，甚至导致系统崩溃。该病毒还会对微软的一个升级网站进行拒绝服务攻击，导致该网站堵塞，使用户无法通过该网站升级系统。

由于蠕虫病毒是计算机病毒的一种，它的传染机理是利用网络进行复制和传播，传染途径是网络和电子邮件，因此“冲击波”病毒属于蠕虫类型的病毒。

(10) C。《中华人民共和国标准化法》是我国标准化工作的基本法，分为五章二十六条，其主要内容是：确定了标准体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。该法将我国标准分为国家标准、行业标准、地方标准、企业标准等 4 个级别。

对于选项 B,“标准”是指为在一定的范围内获得最佳秩序,对活动或其结果规定的共同的和重复使用的规则、导则或特性的文件。该文件经协商一致制定并经一个公认机构的批准。标准应以科学、技术和经验的综合成果为基础,以促进最佳社会效益为目的。

对于选项 A,“标准化”是指为在一定的范围内获得最佳秩序,对实际的或潜在的问题制定共同的和重复使用的规则的活动。它包括制定、发布及实施标准的过程。标准化的重要意义是改进产品、过程和服务的适用性,防止贸易壁垒,促进技术合作。

(11) B。商业秘密权是公民、企业、单位、组织享有的知识产权。商业秘密的构成条件是有关信息具有经济价值、相对秘密性,有关信息的控制人采取了一定的合理保密措施。商业秘密的保护范围包括所有技术信息、经营信息,以及其他与商业竞争有关的信息。商业秘密的侵权行为包括:①以不正当手段获取、披露、使用、允许他人使用权利人的商业秘密;②违反权利人的要求披露、使用、允许他人使用权利人的商业秘密;③对他人的上述行为,第三人明知或者应知,但是仍然接受他人的侵权行为结果,获取、披露、使用、允许他人使用权利人的商业秘密。我国保护商业秘密的法律依据是《反不正当竞争法》。

本案例中,该企业开发设计的应用软件的核心程序设计技巧和算法具有相对秘密性,且企业已对开发人员提出了保密要求,但该开发人员以论文形式向社会披露核心程序设计技巧和算法,因此该开发人员的行为侵犯了企业的商业秘密权。

(12) C。声音信号属于模拟信号,将其转换成为数字信号需经过采样、量化、编码等工作。若量化后的每个声音样本用 2 个字节(16 位)表示,则量化分辨率是 $1/2^{16} = 1/65\,536$ 。

(13) B; (14) A。若每个像素具有 $n=8$ 位的颜色深度,则可表示 $2^n = 2^8 = 256$ 种不同的颜色。若一幅图像具有 640×480 个像素点,每个像素具有 8 位的颜色深度,则未进行压缩时其图像数据所占用的存储空间为 $(640 \times 480 \times 8) \text{ bit} = (640 \times 480) \text{ Byte}$ 。经 5:1 压缩后,图像数据所占用的存储空间为 $\frac{640 \times 480}{5} \text{ Byte} = 61\,440 \text{ Byte}$ 。

(15) A; (16) D。常见的软件开发模型有瀑布模型、演化模型、螺旋模型和喷泉模型等。其中,瀑布模型是一种将按软件生命周期划分为制定计划、需求分析、软件设计、程序编写、软件测试和运行维护六个基本活动,并且规定了它们自上而下、相互衔接的固定次序的系统开发方法。瀑布模型强调文档的作用,并要求每个阶段都要仔细验证,它适用于需求明确或很少变更的项目。

演化模型主要针对事先不能完整定义需求的软件开发项目。根据用户的需求,首先开发核心系统。当该核心系统投入运行后,用户试用并有效地提出反馈。开发人员根据用户的反馈,实施开发的迭代过程。每一次迭代过程由需求、设计、编码、测试和集成等阶段组成,为整个系统增加一个可定义的、可管理的子集。也可将该模型看做是重复执行的多个“瀑布模型”。

螺旋模型是指将瀑布模型和快速原型模型结合起来,强调风险分析的一种开发模型。

喷泉模型基于对象驱动,其开发过程具有迭代性和无间隙性,主要用于描述面向对象的开发过程。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,但不是对先前工作结果的本质性改动。无间隙是指在开发活动(如分析、设计、编程)之间不存在明显的边界,而允许各开发活动交叉、迭代地进行。

常见的几种系统开发模型的优缺点见表 7-5。

表 7-5 常见的几种系统开发模型的优缺点

模 型	优 点	缺 点
瀑布模型	文档驱动	系统可能不满足客户的需求
快速原型模型	关注满足客户需求	可能导致系统设计差,效率低,难以维护
演化模型	开发早期反馈及时,易于维护	需要开放式体系结构,可能导致设计差,效率低
螺旋模型	风险驱动	风险分析人员需要有经验且经过充分训练
喷泉模型	对象驱动,使开发过程具有迭代性和无间隙性	

(17) B。软件能力成熟度模型(CMM)是对软件组织进化阶段的描述。随着定义、实施、测量、控制和改进其软件过程,软件组织的能力逐步前进。CMM将软件过程的成熟度自低到高依次划分为初始级、可重复级、已定义级、已管理级和优化级五个等级,每个等级的特点及关键过程域详见表7-6。

表 7-6 CMM 模型要点概述

级别	描 述	特 点	关键过程域
第一级	初始级	软件过程是无序的,几乎没有明确定义的步骤,成功完全依赖个人努力和英雄式的核心任务;企业一般不具备稳定的软件开发与维护环境,常在遇到问题的时候就放弃原定的计划,而只专注于编程与测试	
第二级	可重复级	在这一级别上,建立了基本的项目管理过程的政策和管理规程来跟踪成本、进度和机能,制定了必要的过程纪律,并基于以往的项目经验来计划与管理新的项目	需求管理、软件配置管理、软件子合同管理等
第三级	已定义级	管理和工程的软件过程已经文档化、标准化,并综合成整个软件开发组织的标准软件过程。所有的项目都采用根据实际情况修改后得到的标准软件过程来发展和维护软件	组织过程定义、集成软件管理、软件产品工程等
第四级	已管理级	在这一级别上,制定了软件工程和产品质量的详细度量标准,使用定量分析来不断地改进和管理软件过程。软件过程和产品的质量都被开发组织的成员所理解和控制,因此软件产品具有可预期的高质量	定量的过程管理、软件质量管理
第五级	优化级	通过来自过程质量反馈和来自新观念、新技术的反馈使过程能持续不断地改进。可见整个企业将会把重点放在对过程进行不断的优化上。企业会采取方法主动找出过程的弱点与优点,以达到预防缺陷的目标	缺陷预防、技术变更管理、过程变更管理

查表 7-6 可知,从可重复级别开始,就要求企业建立基本的项目管理过程的政策和管理规程,使项目管理工作有章可循。一个可管理的过程则是一个可重复的过程,一个可重复的过程则能使项目逐渐演化、成熟。从管理角度看,这是一个按计划执行的、阶段可控的软件开发过程。

(18) D。软件测试通常分为单元测试、组装测试、确认测试和系统测试四个阶段。表 7-7 给出了这四个阶段的主要工作任务和测试依据。

表 7-7 软件测试各阶段的主要任务及依据

阶段	主要任务	测试依据
单元测试	对软件设计的最小单位——模块进行正确性检验的测试	详细设计说明书和源程序
组装测试	也称为集成测试,它是在把模块按照设计要求组装起来的同时进行的测试,主要目的是发现与接口有关的错误	概要设计说明书
确认测试	检验软件的功能和性能及其他特性是否满足需求规格说明书中确定的各种需求,以及软件配置是否完全、正确	需求规格说明书和合同书
系统测试	把通过确认测试的软件作为整个基于计算机系统的一个元素,与计算机硬件、外设、某些支持软件、数据和人员等其他系统元素结合在一起,在实际运行环境下,对计算机系统进行一系列的集成测试和确认测试	

确认测试包括有效性测试和软件配置审查。有效性测试是指在模拟的环境下,运用黑盒测试方法验证所测软件是否满足需求规格说明书列出的要求。在有效性测试中除考虑功能、性能以外,还需检验可移植性、可靠性、兼容性、用户界面及系统所提供的文档资料是否符合要求等内容。软件配置审查的目的在于确保已开发软件的所有文档资料均已编写齐全,足以支持投入运行以后的软件维护工作。

(19) A。面向对象的设计原则之一是:针对接口编程,而不是针对实现编程。遵循这一原则的好处有:① 客户不必知道其使用对象的具体所属类;② 客户无需知道特定类,只需知道他们所期望的接口;③ 一个对象可以很容易地被实现了相同接口的另一个对象所替换;④ 对象间的连接不必硬绑定到一个具体类的对象上,灵活性高;⑤ 增加了重用的可能性,提高了对象组合的机率;⑥ 松散耦合。

(20) B; (21) D。本试题在多道系统中的三个任务不仅要竞争使用 CPU,还要并行使用 I/O 设备。由于本试题为了突出进程对 CPU 和 I/O 设备的使用,弱化了作业调度的处

理,因此题目中已假设这三个任务都已经装入主存,从这一点看,降低了本题的分析难度。分析过程如图7-4所示,图中水平粗实线表示某任务实际执行过程所占用的时间,水平虚线表示该任务等待过程所占用的时间。

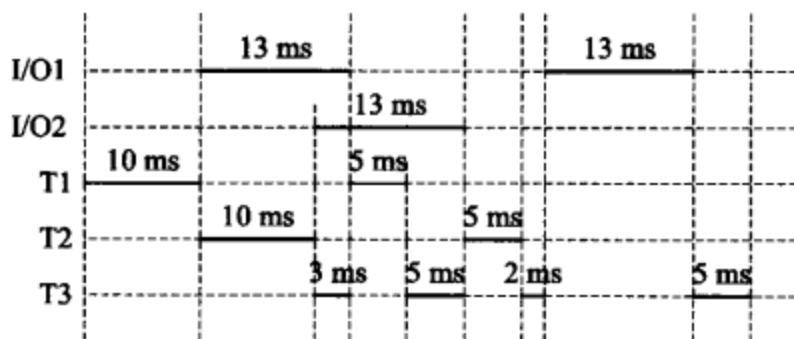


图 7-4 任务运行示意图

如果操作系统的开销忽略不计,设这3个任务从同时启动到全部结束的总时间为 t_1 ,则

$$t_1 = (10 + 13 + 5 + 5 + 5 + 2 + 13 + 5) \text{ ms} = 58 \text{ ms}$$

设CPU的空闲时间为 t_2 ,仔细分析任务运行示意图可知,CPU的空闲时间为任务T3进行I/O操作的时间,即 $t_2 = 13 \text{ ms}$ 。

(22) A。操作系统的分区分配存储管理方案,是一种可用于多道程序的较简单的存储管理方式,又可进一步分为固定分区式(将内存用户区划分成若干个固定大小的区域,每个区域中驻留一道程序)和可变分区(又称为动态分区,根据用户程序的大小,动态地对内存进行划分,各分区的大小是不定的,其数目也是可变的)。由于分区式分配方案要求每个作业占用一个连续的内存空间,因此可能产生多个碎片。该分配方式的特点是:在系统进行初始化的时候就已经将主存储空间划分成大小相等或不等的块,并且这些块的大小在此后是不可以改变的。

请求分页存储管理方案是在分页系统的基础上,增加了请求调页的功能、页面置换等功能所形成的页式虚拟存储系统。它允许只装入若干页(而非全部程序)的用户程序和数据即可启动运行,以后再通过调页功能,陆续地把即将运行的页面调入内存(即不要求作业程序连续存放),同时把暂不运行的页面换出到外存,置换时以页面为单位。该存储管理方案的特点是:主存储空间和程序按固定大小单位进行分割,当一个作业的程序地址空间大于主存可以使用的空间时也可以执行,可有效解决存储空间中的“碎片”问题。

覆盖技术是指一个程序的若干程序段或几个程序的某些部分共享某一个存储空间。它要求用户清楚地了解程序的结构,并指定各程序段调入内存的先后次序(即编程时必须划分程序模块和确定程序模块之间的调用关系)。可见,覆盖管理的目标是逻辑扩充内存,以缓解大作业和小内存之间的矛盾。

(23) C。本试题4个选项逻辑表达式的真值表及其各自对应的波形图如表7-8和表7-9所示。查表可知,与该逻辑门波形图相对应的 $F(A, B)$ 表达式是 $F = A \oplus B$ 。

表 7-8 $A \oplus B$ 和 $A \cdot B$ 的真值表及波形图

$F = A \oplus B$			$F = A \cdot B$		
A	B	F	A	B	F
0	0	0	0	0	0
1	1	0	1	1	1
0	1	1	0	1	0
1	0	1	1	0	0

表 7-9 $A + B$ 和 $A \cdot \bar{B}$ 的真值表及波形图

$F = A + B$			$F = A \cdot \bar{B}$		
A	B	F	A	B	F
0	0	0	0	0	0
1	1	1	1	1	0
0	1	1	0	1	0
1	0	1	1	0	1

(24) C。一个 4 位的二进制计数器，其最大值状态“1111”所能表达的十进制数是 15。计数器达到最大计数状态时，当有下一个时钟脉冲到达时，计数器将从新一轮 0000 状态计数。由此可知，当 4 位的二进制计数器出现第 2 次“0000”状态时，表示已有 16 个时钟脉冲到达。第 17 个时钟脉冲到达后，计数器状态为“0001”，依此类推，经过 25 个时钟脉冲后，该计数器的状态为 1001。

(25) D。稳压电路按稳压元器件与负载的连接关系分为串联型稳压电路和并联型稳压电路。其中，简易的并联型稳压电路如图 7-5 所示，图中稳压二极管与限流电阻呈串联连接关系，与负载电阻呈并联连接关系。

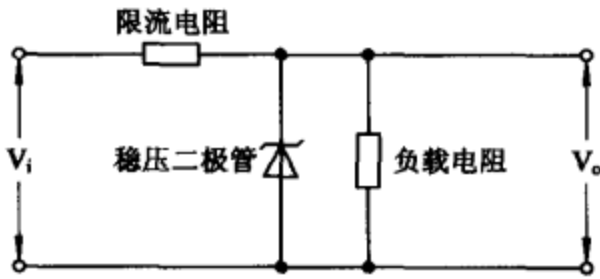


图 7-5 并联型稳压电路

(26) B。RISC 是精简指令集计算机的英文缩写，它是在继承复杂指令集计算机(CISC)技术并克服其缺点的基础上产生并发展的，RISC 与 CISC 在特点上的区别见表 7-10。

表 7-10 RISC 与 CISC 的特点表

类别	RISC	CISC
指令系统	指令长度固定，指令种类少(少于 100 种)	指令数量很多
执行时间	选取使用频率较高的一些简单指令，且指令执行时间较短	有些指令执行时间很长，例如整块的存储器内容复制
编码长度	固定，通常为 4 个字节	可变，1~15 字节
寻址方式	种类少，即简单寻址	种类丰富
操作	设置大量通用寄存器，访问存储器指令简单，只能对寄存器进行算术和逻辑操作	可以对存储器和寄存器进行算术和逻辑操作
编译	采用优化编译技术，生成高效的目标代码程序	难以用优化编译器生成高效的目标代码程序

(27) C。32 位微处理器是指采用 32 位的地址和数据总线的微处理器，其地址空间可达 $2^{32}=4\text{ GB}$ 。32 位/16 位/8 位微处理器体系结构中字、半字、字节的位长度见表 7-11。

表 7-11 不同微处理器体系结构中字、半字、字节的位长度

类型	32 位微处理器体系结构	16 位微处理器体系结构	8 位微处理器体系结构
字	32	16	16
半字	16	16	16
字节	8	8	8

(28) B。当系统传送一个 32 位字时，数据量 $D=32\text{ bit}$ 。由于系统时钟频率 $f=500\text{ MHz}$ ，则系统时钟周期 $T=\frac{1}{f}=\frac{1}{500\times10^6\text{ Hz}}=2\times10^{-9}\text{ s}$ 。设 32 位总线系统的数据传送速率 $S=\frac{D}{t}$ ，由于传送一个 32 位字需要 5 个时钟周期，即 $t=5T$ ，则有

$$S = \frac{D}{t} = \frac{D}{5T} = \frac{32 \text{ bit}}{5 \times 2 \times 10^{-9} \text{ s}} = 3.2 \times 10^9 \text{ bit/s} = \frac{3.2 \times 10^9}{8} \text{ Byte/s} = 400 \text{ MB/s}$$

(29) B。存储管理单元(MMU)在 CPU 和物理内存之间进行地址转换,其主要完成的工作有:① 将地址从虚拟(逻辑)地址空间映射到物理地址空间,即 MMU 实现虚拟地址到物理地址的转换(也将这一转换过程称为内存映射);② 存储器访问权限的控制;③ 设置虚拟存储空间的缓冲特性。

选项 A 的“任务控制块(TCB)”是操作系统中用来描述和管理一个任务的数据结构。

选项 C 的“Cache”位于主存和嵌入式微处理器内核之间,用于减小主存或辅助存储器对微处理器内核造成的存储器访问瓶颈,使处理速度更快,实时性更强。

选项的“DMA”是 Direct Memory Access 的简称,即直接内存访问,是一种数据传输模式。DMA 方式下由于不直接访问计算机的 CPU,而直接在 RAM 与设备之间传输,因而大大提高了数据传输速度。

(30) B。高可靠性和安全性是嵌入式系统的一大特点,也是嵌入式系统最重要、最突出的基本要求。可靠性是一个嵌入式系统正常工作的能力,一般用平均故障间隔时间(MTBF)来度量。

系统的可维护性是指系统失效后在固定时间内可修复到规定功能的能力,通常用平均修复时间(MTTR)来度量。

(31) C。高速缓存(Cache)是一种容量小、速度快的存储器阵列,它位于主存和嵌入式微处理器内核之间,用于存放最近一段时间微处理器使用最多的程序代码和数据。Cache 提高了内存访问的可变性。DRAM 是一种 RAM 类的主存储器,用于存储用户的程序和数据;Flash 存储器、磁盘也主要用于存储用户的程序和数据。

(32) B。除了 I/O 设备本身的性能外,系统总线的传输速率是影响嵌入式系统 I/O 数据传输速度的主要因素。

(33) C; (34) B。RS-422 标准是基于 RS-232 标准发展的,它定义了一种平衡通信接口,将传输速率提高到 10 Mb/s,允许在一条平衡总线上连接最多 10 个接收器。RS-422 标准的数据信号采用差分传输方式(即逻辑信号用两条线的电平差表示),使用一对双绞线进行信号传输。使用双线传输的主要优点是增强了信号的抗干扰能力。

(35) D。解答本试题时建议从第 2 个信号周期开始分析,且先分析差分曼彻斯特编码的比特串值,差分曼彻斯特编码规则是:每比特的中间有一个电平跳变,但利用每个码元的开始是否有跳变来表示“0”或“1”,如有跳变则表示“0”,无跳变则表示“1”,可简记为“0 跳变,1 保持”。

由此规则分析试题中给出的信号波形可得,实际传送的比特串为 011010011。

曼彻斯特编码规则是:每比特的中间有一个电平跳变,从高到低的跳变表示比特“0”,从低到高的跳变表示比特“1”,可简记为“010, 101”,其中带下划线的 0、1 表示实际的比特。由此规则分析试题中给出的信号波形,所得的实际传送的比特串也为 011010011。

(36) D。PCI 总线是当前最流行的系统总线之一,是地址、数据多路复用的高性能 32 位和 64 位总线,即从数据宽度看,PCI 定义了 32 位数据总线,且可扩展到 64 位。从总线速度上分为 33 MHz 和 66 MHz 两种。PCI 总线是一种独立设计的总线,它的性能不受 CPU 类型的影响。与 ISA 总线的区别之处是,PCI 总线的地址总线与数据总线是分时复用

的,支持即插即用、中断共享、数据突发读/写操作等功能。在普通PC机中允许同时使用PCI总线和ISA总线。

(37) B。IEEE 802.11b是目前应用最广的一种WLAN标准,使用开放的2.4 GHz频段直接序列扩频,最大数据传输速率为11 Mb/s。

IEEE 802.11a、IEEE 802.11b和IEEE 802.11g三种无线局域网标准的异同点见表7-12。

表 7-12 IEEE 802.11 的三个子标准异同点对比表

比较项目	IEEE 802.11a	IEEE 802.11b	IEEE 802.11g
工作频段	5 GHz	2.4 GHz	2.4 GHz
最高速率	可达 54 Mb/s	11 Mb/s	可提升至 54 Mb/s
通信技术	OFDM	DSSS CCK 调制	OFDM
业务	数据、语音、图像	数据、图像	数据、语音、图像
兼容性	与 IEEE 802.11b 标准产品不兼容	与 IEEE 802.11a 标准产品不兼容	与现有 IEEE 802.11b 标准兼容

(38) D。电路板的设计按以下三个步骤依次进行,即设计电路原理图(SCH)→生成网络表(NET)→设计印制电路板(PCB)。自动布线工作只是印制电路板设计流程中的一部分工作。

(39) B。现代电子设计方法包含了可测试设计,测试的任务就是确认IC芯片内部是否有隐藏的故障。芯片测试通常分为样品测试和产品测试。JTAG接口是IC芯片测试的标准接口。

选项A的“BIST(内建测试系统)”是SOC片上系统的重要结构之一。

选项C的“UART”是通用异步收发器的英文缩写。

选项D的“USB”是具有热插拔特性的通用串行总线的英文缩写。

(40) B。利用多层印制电路板(4层或者4层以上)进行PCB布线可以大大减小电路中信号回路的面积,因此比双面板更适合于高速PCB布线。但制作多层印制电路板的工艺较复杂。

(41) C。数字电路(或者集成电路)的电磁兼容性(EMC)设计在进行器件选择时,通常遵循以下原则:①在满足产品技术指标的前提下,尽量选择低速时钟的IC;②所选IC的电源及地的引脚较近;③所选IC有多个电源及地线引脚;④所选IC输出电压波动性小,输出级驱动能力不超过实际应用的要求;⑤所选IC的电源瞬态电流(穿透电流)低;⑥所选IC的输入电容尽量小;⑦所选IC具有可控开关速率;⑧所选IC地线反射较低;⑨建议直接在PCB上焊接表贴芯片,具有较短引线 and 体积较小的IC芯片更好等等。

在EMC电路技术方面,通常遵循以下原则:①对输入和按键采用电平检测(而非边沿检测);②降低负载电容,以使靠近输出端的集电极开路驱动器便于上拉,电阻值尽量大;③使用前沿速率尽可能慢且平滑的数字信号(不超过失真极限);④在PCB样板上,允许对信号边沿速度或带宽进行控制(例如,在驱动端使用软铁氧体磁珠或串联电阻);⑤微处理器散热片与芯片之间通过导热材料隔离,并在处理器周围多点射频接地;⑥不能在看门

狗或电源监视电路上使用可编程器件, 电路中尽量配置一只高质量的看门狗; ⑦ 电源的监视电路需对电源中断、跌落、浪涌和瞬态干扰有抵抗能力; ⑧ 当逻辑信号沿的上升/下降时间比信号在 PCB 走线中传输一个来回的时间短时, 应采用传输线技术等等。

(42) C. 嵌入式系统由硬件部分和软件部分构成。通常将嵌入式软件分为嵌入式系统软件、嵌入式应用软件和嵌入式支撑软件三大类。每类软件的功能及应用例子见表 7-13。

表 7-13 嵌入式软件分类表

软件类型	功 能	例 子
系统软件	控制和管理嵌入式系统资源, 为嵌入式应用提供支持的各种软件	嵌入式操作系统、设备驱动程序、嵌入式中间件等
应用软件	定义嵌入式设备的主要功能和用途, 并负责与用户进行交互	MP3 播放软件、手机软件、路由策略软件、恒温控制软件等
支撑软件	辅助软件开发的工具软件	在线仿真工具、FPGA 编程软件、交叉编译器、源程序模拟器等

(43) A. 硬件抽象层(HAL)也称为板级支持包(BSP), 位于嵌入式系统硬件层和系统软件层之间, 将系统上层软件与底层硬件分离开来, 使系统的底层驱动程序与硬件无关。它主要包括引导加载程序 BootLoader 和设备驱动程序两个方面的内容, 具有以下两个特点:

① 硬件相关性。因为嵌入式实时系统的硬件环境具有应用相关性, 而作为上层软件与硬件平台之间的接口, BSP 需为操作系统提供操作和控制具体硬件的方法;

② 操作系统相关性。不同的操作系统具有各自的软件层次结构, 因此不同操作系统具有特定的硬件接口形式。

(44) D. 任务是相应任务块程序关于某数据集合的可并发的一次运行活动, 通常作为占有资源的最基本单位。通常在嵌入式实时操作系统中, 任务总共有三个状态, 分别是阻塞状态、就绪状态和运行状态。在任何时候, 一个任务只会处于其中的某一个状态。

其中, ① 阻塞状态是指任务因为正在等待某种事件的发生而暂时不能运行。例如, 它正在等待某个 I/O 操作的完成, 或者它同某个任务之间存在着同步关系, 正在等待该任务给它发信号。此时, 即使 CPU 已经空闲下来了, 它也还是不能运行。处于挂起状态的任务可以被某些系统调用或事件激发而转到就绪状态。

② 就绪状态是指任务已经具备了运行的条件, 但是由于 CPU 正忙, 正在运行其他的任务, 因而暂时不能运行的状态。当一个任务被新创建时就处于就绪状态。同样当一个任务被唤醒, 意味着该任务将进入就绪状态。当任务调度器选中某一任务去运行时, 该任务将进入运行状态。

③ 运行状态表示此任务正在占有 CPU, 并在 CPU 上运行。处于就绪状态的任务只要把 CPU 控制权分配给它(任务调度), 它就能够立刻执行。当一个任务运行过程中需要等待某个事件时, 将由运行状态转移到等待状态。当有比正在运行的任务的高优先级事件发生时, 运行状态将被高优先级的任务抢占, 即高优先级中断可以使正在执行的任务转移到就绪状态。

(45) B。对于实时操作系统(RTOS)的任务调度器,任务之间的实时性是最重要的调度目标,即要让每个任务都在最终时间期限之前完成。大多数 RTOS 调度算法都采用基于优先级的可抢占式(可剥夺式)算法。

单调速率(RMS)调度算法、最早期限优先(EDF)调度算法是常用的确定任务优先级的算法。其中,RMS 调度算法是一种静态优先级调度算法,任务的周期越短,则优先级越高,反之优先级越低。EDF 调度算法是一种动态优先级调度算法,它能根据需要动态地修改各个任务的优先级。

(46) A。时间片轮转(RR)算法的一个默认前提是:位于就绪队列当中的各个任务是同等重要的,因此该算法把这些任务按照先来先服务的原则排成一个队列,并轮流执行相同长度的时间片。公平性和活动性是其两大优点。

先来先服务算法(FCFS)按照任务到来的先后次序进行调度,是一种不可抢占的调度方式,当长周期高优先级的任务占用 CPU 资源时,将引起短周期任务出现“饥饿”现象。

可抢占的短作业优先算法也称为最短剩余时间优先算法(SRTF)可能会使最早到达而优先级最低的任务出现“饥饿”现象。

静态优先级算法最大的缺点是:当高优先级的任务一直占用 CPU 资源时,那些低优先级的任务可能会长时间得不到 CPU 资源而一直处于“饥饿”状态。

(47) B。在实时操作系统中,任务间通信是指任务之间为了协调工作,需要相互交换数据和控制信息,可分为低级通信和高级通信两种类型。PV 操作等信号量机制和信号机制属于低级通信方式;消息传递、管道、共享内存属于高级通信方式。

任务间通信又可分为直接通信和间接通信两种类型。send、receive 原语属于直接通信方式;邮箱、消息队列属于间接通信方式。

(48) B。信号量 S 表示资源的实体,它是一个与队列有关的整型变量,其值仅由 P、V 操作来完成。当 $S=n>0$ 时,表示队列中有 n 个资源可用;当 $S=0$ 时,表示队列中无资源可用;当 $S=n<0$ 时,表示进入等待队列中的任务个数有 n 的绝对值 $|n|$ 个。本试题中,信号量 S 的当前值为 -1,则表示等待信号量 S 的任务个数为 1 个。

(49) A。在实时操作系统中,多个任务并发执行,一个任务要等待其合作伙伴发来消息,或建立某个条件后再向前执行,这种制约性合作关系被称为任务的同步。

多个并发任务由于不能同时使用一临界资源,只能在一个任务使用完该资源后,另一任务才能使用的现象称为任务的互斥。

(50) C。在一些低端的、比较简单的嵌入式应用中,基于成本等因素的考虑,通常采用低速的 CPU。低速 CPU 不具备 MMU 的功能,不需要内核提供内存保护机制。

在嵌入式操作系统中,常见的存储管理方案可分为实模式方案和保护模式方案两种类型。在实模式存储管理方式中,系统将关闭 MMU 或根本没有 MMU,不划分“系统空间”和“用户空间”,整个系统只有一个地址空间(即物理内存地址空间)。在保护模式方案中,处理器必须要有 MMU 硬件并启用,内核空间和用户空间必须是两个相互独立的地址空间,具有存储保护功能。也可以没有 MMU,不划分“系统空间”和“用户空间”,整个系统只有一个地址空间(即物理内存地址空间)。

在虚存系统中,任务拥有的最大编址空间受处理器的地址结构、可用的辅助存储器容量等因素的限制。

(51) B。进行存储分配时,页式存储管理为每个作业建立一张页表,指出逻辑地址中页号与主存中块号的对应关系。然后,借助于硬件的地址转换机构,在作业执行过程中按页动态定位。每执行一条指令时,按逻辑地址中的页号查页表,得到对应的块号,根据关系式:绝对地址=块号×块长+单元号,计算出欲访问的主存单元的地址。由此可知,页面是由操作系统所感知的。

(52) A。文件是一个具有符号名的相关元素的有序序列。操作系统中负责管理和存取文件信息的软件机构称为文件系统,其主要功能是实现对文件的按名存取。

虚拟存储的基本思想是以牺牲 CPU 的工作时间以及内、外存的交换时间为代价,为用户提供一个足够大的地址空间。

高速缓存(Cache)位于主存与微处理器之间,用于提高微处理器对外存的读写速度。

(53) D。FAT(File Allocation Table)是一种适合小卷集与双重引导需要的文件系统。采用 FAT 文件系统格式化的卷以簇的形式进行分配。默认的簇的大小由卷的大小决定。对于 FAT 文件系统,簇数目必须为 512 个字节到 65 536 个字节之间的 2 的次幂。表 7-14 列出 FAT16 文件系统缺省的簇空间大小。

表 7-14 FAT16 文件系统默认的簇空间大小

卷空间的大小	每个簇的扇区	簇空间大小
0~32 MB	1	512 个字节
33~64 MB	2	1 KB
65~128 MB	4	2 KB
129~255 MB	8	4 KB
256~511 MB	16	8 KB
512~1023 MB	32	16 KB
1024~2047 MB	64	32 KB
2048~4096 MB	128	64 KB

由表 7-14 表可知,若每个簇的大小是 2 KB,那么它所能表示的最大磁盘分区容量为 $2^{16} \times 2 \text{ KB} = 2^7 \times 2^{10} \text{ KB} = 128 \text{ MB}$ 。

(54) D。嵌入式应用系统通常是一种定制系统,不同应用环境对 GUI 的需求也各不相同。有的系统只要求一些简单的图形功能,而有些系统要求完备的 GUI 支持。因此,GUI 也必须是可定制的。嵌入式系统或者实时系统对 GUI 的基本要求通常包括:① 轻型、运行时占用的系统资源少;② 高可靠性;③ 高性能;④ 模块化结构,可配置可定制。华丽的外观,复杂的图形算法并不是嵌入式系统对 GUI 最主要考虑的问题。

(55) C。在宿主机与目标机之间必须建立物理连接和逻辑连接。物理连接是指宿主机与目标机之间通过物理线路将其一定的物理端口连接在一起,其连接方式主要有串口、以太网接口和 OCD(On Chip Debug)三种。逻辑连接是指宿主机与目标机之间按某种通信协议建立起来的虚连接。

在嵌入式系统开发中,通常采用交叉编译器将宿主机上编写的高级语言程序编译成可以在目标机上运行的二进制代码。

在嵌入式系统中,由于目标机的资源有限,调试器与被调试程序通常运行在不同的机器上。调试器主要运行在宿主机上,而被调试程序则运行在目标机上。

(56) B。汇编语言是为特定的计算机系统设计的面向机器的符号化程序设计语言。由于计算机不能直接识别和运行符号语言程序,因此需要用专门的翻译器——汇编程序进行翻译。由此可见,汇编语言程序不能直接在计算机上运行。

汇编语言源程序通常由指令语句、伪指令语句和宏指令语句组成。

其中,① 指令语句经汇编后要产生相应的能被 CPU 直接识别并执行的机器代码,其所指示的操作是在程序运行时完成的。

② 伪指令语句指示汇编程序在对源程序进行汇编时完成某些工作,进行汇编时不产生机器代码,其所指示的操作是在源程序被汇编时完成的。通常汇编语言设有常数定义伪指令、存储定义伪指令、汇编控制伪指令、开始伪指令、结束伪指令等。其中,常数定义伪指令可用于定义符号常量;汇编控制伪指令用于控制汇编程序的执行流程。

③ 宏指令语句就是宏引用,允许用户多次重复使用某一程序段。

(57) A。栈是一种先进后出的数据结构,栈有一个存储区和一个栈顶指针。栈顶指针指向堆栈中第一个可用的数据项(被称为栈顶)。用户可以在栈顶上方向栈中加入数据,这个操作被称为压栈(push)。压栈以后,栈顶自动变成新加入数据项的位置,栈顶指针也随之修改。用户也可以从堆栈中取走栈顶,称为弹出栈(pop)。弹出栈后,栈顶下的一个元素变成栈顶,栈顶指针随之修改。

本试题中,由于 `eax` 是一个 32 位寄存器,其值将占用 4 字节存储空间,因此执行“push `eax`”指令后,栈顶指针寄存器的当前值将被修改为 $(0x00FFFFE8-4)=0x00FFFFE4$ 。

(58) C。字符数组是每个元素存放一个字符型数据的数组。C 语言中约定用“\0”作为字符串的结束标志,它占内存空间,但不计入字符串的长度。对数组 `array` 赋初值: `char array[]="China"`,此时系统自动在末尾加入一个“\0”,此时数组 `array` 的长度为 6,所占用的空间为 6 个字节。若定义 `char array[]={'C','h','i','n','a','\0'}`,此时数组 `array` 的长度为 5。

`char array[]="China"`等价于 `char array[]={'C','h','i','n','a','\0'}`。

(59) A。本题的解题关键是理解“++a”,它代表使用该变量时要先将其值加 1,即“先加后用”,而 `a++` 则表示执行后 `a` 的值加 1,即“先用后加”。因此执行 C 程序代码“`int a=1; int b=0; int c=0; int d=(++a)*(c=1);`”后,`a` 的值由 1 变为 2,`b` 的值为 0,`c` 的值由 0 变为 1,而 `d=2×1=2`。

(60) A。定义函数时函数名后括号中的变量名称为形式参数。调用函数时函数名后括号中的表达式称为实际参数。形参在未出现函数调用时,并不占内存中的存储单元。只有在发生函数调用时,函数中的形参才被分配内存单元。在调用结束后,形参所占的内存单元也被释放。因此选项 A 说法正确。

C 语言规定,实参对形参的数据传递是“值传递”,即单向传递(实参的值可以传给形参,形参的值不能传给实参)。在内存中,实参单元与形参单元是不同的单元。在被调用的函数当中,函数返回值将影响主调函数中相关变量的值。由此可见,选项 B 说法错误。

通常认为数组名是一个地址常量,当数组名作为形参时,则传递的是数组的起始地址,而不是每个数组元素的值。由此可见,选项 D 说法错误。

操作系统内存的栈空间用于保存运行的上下文、函数调用时的局部变量和形参。因此选项 C 说法错误。

(61) D。嵌入式应用软件的移植指的是把应用软件从一个嵌入式操作系统平台移植到另一个操作系统平台。为了提高嵌入式应用软件的移植性，在软件设计时通常遵循如表 7-15 所示的一些开发原则。

表 7-15 嵌入式应用软件开发的一些通用原则

原 则	描 述
① 在软件设计上，尽量采用层次化设计和模块化设计	在软件纵向结构上，下层为上层提供服务，上层去调用下层提供的服务。每一个层次都应该定义清晰的接口和功能，但分层的数量要合适。在整体软件的设计上，可同一层的软件结构上要尽量形成模块化，且软件模块之间是相互独立的，一个模块的实现不依赖于其他模块的实现
② 在软件体系结构上，尽量在操作系统和应用软件之间引入一个虚拟机层（也称为操作系统抽象层）	把一些通用的、共性的操作系统的 API 接口函数封装起来，在编写一个应用程序时，不是直接去调用实际操作系统的 API，而是使用虚拟层所提供的 API。在新的操作系统平台上仅需实现此虚拟层，而无需修改其他的代码
③ 将不可移植的部分局域化	把不可移植的代码通过宏定义和函数的形式，分类集中于某几个特定的文件之中。在以后的移植过程中，既有利于迅速地对需要修改的代码进行定位，又可方便地进行修改
④ 在数据类型上，用宏定义的方式定义一组可移植的数据类型	在应用程序的内部，只使用宏定义的数据类型，而不使用 C 语言的数据类型。这是由于 C 语言的数据类型与机器的字长和编译器有关。例如，可以用 INT32U 来表示无符号的 32 位整型数据，对于实际的编译器，可以定义为 <code>#define INT32U unsigned int</code>
⑤ 在功能服务的调用上，要尽量使用可移植的函数	如标准的 C 语言函数或自己编写的函数，尽量不要使用依赖于特定操作系统的 API 函数
⑥ 提高代码的可重用性	例如，可以更好地抽象软件的函数，使之更加模块化，功能更专一，接口更简洁明了

(62) D。在大型实时系统开发项目中，体系结构设计产生的输入信号预处理、主控制过程、网络接口等部件设计均需考虑非功能需求的可靠性。

(63) A。测试是保证软件质量的重要手段。根据国家标准 GB 8566—88《计算机软件开发规范》的规定，单元测试是根据详细设计阶段给出的规格说明书在编码阶段完成的测试

工作,其典型输入是程序代码;集成测试的计划是在概要设计阶段制定的,其典型输入是系统设计说明书;系统测试计划应该在需求分析阶段就开始制定,并在设计阶段细化和完善,其典型输入是需求规格说明书和合同书;而验收测试则检测产品是否符合最终用户的需求。软件测试的各个阶段与软件开发阶段的对应关系如图7-6所示。

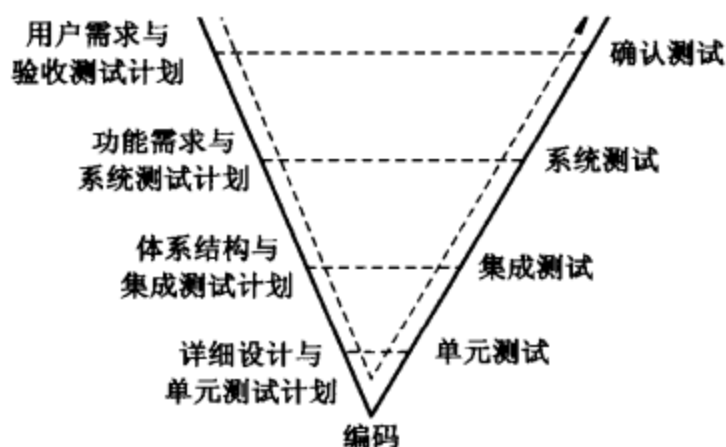


图7-6 各软件开发阶段与软件测试阶段的对应关系

(64) B。系统建模是系统分析阶段的一个重要环节,用于描述所建立系统的软/硬件模型、优化系统描述的过程。系统分析模型应明确体现的因素有:①性能描述,反映的是系统的整体面貌及体系结构,应明确或隐含地说明系统 I/O 及相关的中间状态,以及相互之间的关系;②功能描述,应明确表述系统各项功能特点与系统 I/O 及相关的中间状态之间的关系;③约束条件,应明确规定技术指标的适用范围、系统的工作环境要求及系统性能的缺陷、不足等;④动态模型,是评价系统质量的指标体系,常与价格、速度、字长、可靠性等具体项目有关。

(65) A。嵌入式系统软/硬件协同设计从目标系统构思开始,经过系统需求描述、模块的行为描述、对模块的有效性检查、软/硬件划分、性能评估、硬件综合、接口综合、软件编译、软/硬件集成、软/硬件协同仿真、系统评估与设计验证等各个阶段。

(66)~(70): B、D、C、A、C。

参考译文:嵌入式系统是以应用为中心,以计算机技术为基础,并且软、硬件可裁剪,适用于系统对功能、可靠性、成本、体积和功耗等有严格要求的专用计算机系统。随着集成电路(IC)设计和制造技术的快速发展,CPU 的价格变得越来越便宜。越来越多的消费类电子中使用了嵌入式 CPU,也就成为了嵌入式系统。例如,PDA、手机、POS 机、录像机(VCR)和工业机器人控制等。

(71)~(75): C、D、B、A、C。

参考译文:嵌入式操作系统是在引导装入程序对嵌入式设备进行初始化引导启动后,管理运行在设备上的所有其他程序的软件。嵌入式系统在实现应用程序的同时,能保证软件的存储空间和操作的时间满足一定的要求。嵌入式操作系统都较小,一般仅包含初始引导装入程序、操作系统内核、设备驱动程序和用于存储用户数据的文件系统。嵌入式系统与常见的操作系统在结构上非常相似,区分它们的主要因素有预安装设备、功能有限和仅执行特别设计的任务等。

7.2 下午试题

下午试题 1

请参见本书第 1 章 1.3.4 节。

下午试题 2 至试题 4

请参见本书第 3 章 3.3.1 节~3.3.3 节。

下午试题 5

参见本书第 2 章 2.3.1 节。

欲知
和聲
PDG

第8章 上午试卷公共试题(中级)

8.1 2004年下半年真题链接

8.1.1 上午试题

• 内存按字节编址,地址从 0A4000H 到 0CBFFFH,共有(1)字节。若用存储容量为 32 K×8 bit 的存储器芯片构成该内存,至少需要(2)片。

- (1) A. 80 K B. 96 K C. 160 K D. 192 K
(2) A. 2 B. 5 C. 8 D. 10

• 中断响应时间是指(3)。

- (3) A. 从中断处理开始到中断处理结束所用的时间
B. 从发出中断请求到中断处理结束所用的时间
C. 从发出中断请求到进入中断处理所用的时间
D. 从中断处理结束到再次中断请求的时间

• 若指令流水线把一条指令分为取指、分析和执行三部分,且三部分的时间分别是 $t_{\text{取指}}=2\text{ ns}$, $t_{\text{分析}}=2\text{ ns}$, $t_{\text{执行}}=1\text{ ns}$,则 100 条指令全部执行完毕需(4)ns。

- (4) A. 163 B. 183 C. 193 D. 203

• 在单指令流多数据流计算机(SIMD)中,各处理单元必须(5)。

- (5) A. 以同步方式,在同一时间内执行不同的指令
B. 以同步方式,在同一时间内执行同一条指令
C. 以异步方式,在同一时间内执行不同的指令
D. 以异步方式,在同一时间内执行同一条指令

• 单个磁头在向盘片的磁性涂层上写入数据时,是以(6)方式写入的。

- (6) A. 并行 B. 并-串行 C. 串行 D. 串-并行

• 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,那么主存地址应为(7)位,主存区号应为(8)位。

- (7) A. 16 B. 17 C. 18 D. 19
(8) A. 5 B. 6 C. 7 D. 8

• 软件开发中的瀑布模型典型地刻画了软件生存周期的阶段划分,与其最相适应的软件开发方法是(9)。

- (9) A. 构件化方法 B. 结构化方法
C. 面向对象方法 D. 快速原型方法

• 下述任务中,不属于软件工程需求分析阶段的是(10)。

(10) A. 分析软件系统的数据要求

B. 确定软件系统的功能需求

C. 确定软件系统的性能要求

D. 确定软件系统的运行平台

• 软件设计的主要任务是设计软件的结构、过程和模块,其中软件结构设计的主要任务是要确定(11)。

(11) A. 模块间的操作细节

B. 模块间的相似性

C. 模块间的组成关系

D. 模块的具体功能

• 系统测试是将软件系统与硬件、外设和网络等其他因素结合,对整个软件系统进行测试。(12)不是系统测试的内容。

(12) A. 路径测试

B. 可靠性测试

C. 安装测试

D. 安全测试

• 项目管理工具中,将网络方法用于工作计划安排的评审和检查的是(13)。

(13) A. Gantt 图

B. PERT 网图

C. 因果分析图

D. 流程图

• 在结构化分析方法中,数据字典是重要的文档。对加工的描述是数据字典的组成内容之一,常用的加工描述方法(14)。

(14) A. 只有结构化语言

B. 有结构化语言和判定树

C. 有结构化语言、判定树和判定表

D. 有判定树和判定表

• CMM 模型将软件过程的成熟度分为 5 个等级。在(15)使用定量分析来不断地改进和管理软件过程。

(15) A. 优化级

B. 管理级

C. 定义级

D. 可重复级

• 在面向数据流的设计方法中,一般把数据流图中的数据流划分为(16)两种。

(16) A. 数据流和事务流

B. 变换流和数据流

C. 变换流和事务流

D. 控制流和事务流

• (17)属于第三层 VPN 协议。

(17) A. TCP

B. IPsec

C. PPoE

D. SSL

• 图 8-1 所示的防火墙结构属于(18)。

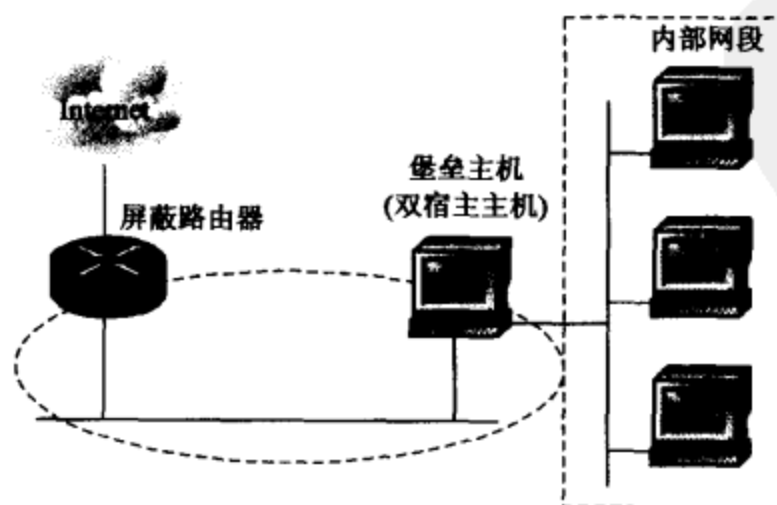


图 8-1 某防火墙拓扑结构图

- (18) A. 简单的双宿主主机结构 B. 单 DMZ 防火墙结构
C. 带有屏蔽路由器的单网段防火墙结构 D. 双 DMZ 防火墙结构

• 电子商务交易必须具备抗抵赖性, 目的在于防止(19)。

- (19) A. 一个实体假装成另一个实体
B. 参与此交易的一方否认曾经发生过此次交易
C. 他人对数据进行非授权的修改、破坏
D. 信息从被监视的通信过程中泄漏出去

• 知识产权一般都具有法定的保护期限, 一旦保护期限届满, 权利将自行终止, 成为社会公众可以自由使用的知识。(20)权受法律保护的期限是不确定的, 一旦为公众所知悉, 即成为公众可以自由使用的知识。

- (20) A. 发明专利 B. 商标
C. 作品发表 D. 商业秘密

• 甲、乙两人在同一时间就同样的发明创造提交了专利申请, 专利局将分别向各申请人通报有关情况, 并提出多种解决这一问题的办法, 不可能采用(21)的办法。

- (21) A. 两申请人作为一件申请的共同申请人
B. 其中一方放弃权利并从另一方得到适当的补偿
C. 两件申请都不授予专利权
D. 两件申请都授予专利权

• 《计算机软件产品开发文件编制指南》(GB 8567—88)是(22)标准。

- (22) A. 强制性国家 B. 推荐性国家
C. 强制性行业 D. 推荐性行业

• 虚拟存储管理系统的基础是程序的(23)理论, 这个理论的基本含义是指程序执行时往往会不均匀地访问主存储器单元。根据这个理论, Denning 提出了工作集理论。工作集是进程运行时被频繁地访问的页面集合。在进程运行时, 如果它的工作集页面都在(24)内, 能够使该进程有效地运行, 否则会出现频繁的页面调入/调出现象。

- (23) A. 全局性 B. 局部性
C. 时间全局性 D. 空间全局性
(24) A. 主存储器 B. 虚拟存储器
C. 辅助存储器 D. U 盘

• 在 UNIX 操作系统中, 若用户键入的命令参数的个数为 1, 则执行 `cat $1` 命令; 若用户键入的命令参数的个数为 2, 则执行 `cat >> $2 < $1` 命令。请将下面所示的 Shell 程序的空缺部分补齐。

```
case(25)in
  1)cat $1;;
  2)cat >> $2 < $1;;
  *)echo 'default... '
esac
```

- (25) A. \$ \$ B. \$ @ C. \$ # D. \$ *

• 以太网 100BASE-TX 标准规定的传输介质是(26)。

(26) ^① A. 3 类 UTP B. 5 类 UTP C. 单模光纤 D. 多模光纤

• 许多网络通信需要进行组播, 以下选项中不采用组播协议的应用是(27)。在 IPv4 中把(28)类地址作为组播地址。

(27) A. VOD B. Netmeeting C. CSCW D. FTP

(28) A. A B. B C. D D. E

• 将双绞线制作成交叉线(一端按 EIA/TIA 568A 线序, 另一端按 EIA/TIA 568B 线序), 该双绞线连接的两个设备可为(29)。

(29) A. 网卡与网卡

B. 网卡与交换机

C. 网卡与集线器

D. 交换机的以太网口与下一级交换机的 UPLINK 口

• 以下不属于中间件技术的是(30)。

(30) A. Java RMI

B. CORBA

C. DCOM

D. Java Applet

• Networks can be interconnected by different devices in the physical layer networks can be connected by (31) or hubs, which just move the bits from one network to an identical network. One layer up we find bridges and switches which operate at data link layer. They can accept (32) examine the MAC address and forward the frames to a different network while doing minor protocol translation in the process in the network layer, we have routers that can connect two networks. If two networks have (33) network layer, the router may be able to translate between the packet formats. In the transport layer we find transport gateway, which can interface between two transport connections. Finally, in the application layer, application gateways translate message (34). As an example, gateways between Internet E-mail and X.400 E-mail must (35) the E-mail message and change various header fields.

(31) A. repeaters B. relays C. packages D. modems

(32) A. frames B. packets C. packages D. cells

(33) A. special B. dependent C. similar D. dissimilar

(34) A. syntax B. semantics C. language D. format

(35) A. analyze B. parse C. delete D. create

• The purpose of the requirements definition phase is to produce a clear, complete, consistent, and testable (36) of the technical requirements for the software product.

During the requirements definition phase, the requirements definition team uses an iterative process to expand a broad statement of the system requirements into a complete and detailed specification of each function that the software must perform and each (37) that it must meet. The starting point is usually a set of high level requirements from the (38) that describe the project or problem.

In either case, the requirements definition team formulates an overall concept for the

^① 注: 试题编号(26)~(40)在原试题中的编号为(61)~(75)。

system and then defines (39) showing how the system will be operated publishes the system and operations concept document and conducts a system concept review(SCR).

Following the SCR, the team derives (40) requirements for the system from the high level requirements and the system and operations concept. Using structured or object-oriented analysis. The team specifies the software functions and algorithms needed to satisfy each detailed requirement.

- | | | | |
|-------------------|---------------|------------------|--------------|
| (36) A. function | B. definition | C. specification | D. statement |
| (37) A. criterion | B. standard | C. model | D. system |
| (38) A. producer | B. customer | C. programmer | D. analyser |
| (39) A. rules | B. principles | C. scenarios | D. scenes |
| (40) A. detailed | B. outlined | C. total | D. complete |

8.1.2 参考答案

表8-1给出本份真题问题(1)~(40)的参考答案,供读者练习时参考,以便查缺补漏。读者可按每空1分的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

表8-1 参考答案及评分标准表

问题号	参考答案	问题号	参考答案
(1)~(5)	C、B、C、D、B	(21)~(25)	D、A、B、A、C
(6)~(10)	C、D、B、B、D	(26)~(30)	B、D、C、A、D
(11)~(15)	C、A、B、C、B	(31)~(35)	A、A、D、A、B
(16)~(20)	C、B、B、B、D	(36)~(40)	C、A、B、A、B

8.1.3 要点解析

(1) C; (2) B。试题(1)要求计算按字节编址的内存从0A4000H到0CBFFFH的地址空间。首先考虑该连续的地址空间应包括0A4000H该字节的地址,因此该地址空间是: $(0CBFFFH+1)-0A4000H=0CC000H-0A4000H$ 。

接着考虑到试题(1)的4个选项的单位均以KB为计算单位,其中 $1KB=2^{10}B=0B10\ 0000\ 0000B$ 。因此对于 $0CC000H-0A4000H$ 算式中只需重点关注高10位的计算,其计算过程如下:

$$\begin{array}{rcl}
 1100\ 1100\ 00 & \leftarrow & 0CC000H\text{的高10位} \\
 -1010\ 0100\ 00 & \leftarrow & 0A4000H\text{的高10位} \\
 \hline
 10\ 1000\ 00 & \leftarrow & \text{计算结果以KB为单位}
 \end{array}$$

计算结果 $0B10\ 1000\ 00 = 2^7 + 2^5 = 128 + 32 = 160$, 因此试题(1)的答案是选项 C。

若要用存储容量为 $32\ K \times 8\ bit$ 的存储器芯片构成该 $160\ KB$ 的内存, 设至少需要存储器芯片的片数为 N 。

$$N = (160 \times 1024 \times 8\ bit) / (32 \times 1024 \times 8\ bit) = 5$$

因此试题(2)的答案是选项 B。

(3) C。中断响应时间是指从发出中断请求到进入中断处理所用的时间; 而从中断处理开始到中断处理结束所用的时间称为中断处理时间, 如图 8-2 所示。

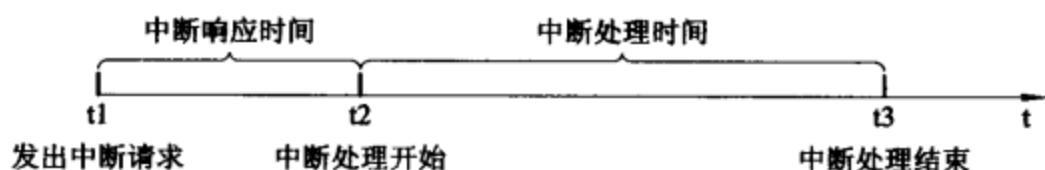


图 8-2 中断响应时间及中断处理时间示意图

(4) D。流水线技术其实质是一种任务的分解技术, 这种技术把一件任务分解为若干顺序执行的子任务, 不同的子任务由不同的执行机构负责执行, 而这些机构可以同时并行工作。在任一时刻, 任一任务只占用其中一个执行机构, 这样就可以实现多个任务的重叠执行, 以提高工作效率。若指令流水线把一条指令分为取指、分析和执行三部分, 且三部分的时间分别是 $t_{取指} = 2\ ns$, $t_{分析} = 2\ ns$, $t_{执行} = 1\ ns$, 则该流水线的操作周期应取其中时间最长的 $t_{取指} = 2\ ns$ 或 $t_{分析} = 2\ ns$ 操作部分。

在此指令流水线上执行第 1 条指令是流水线的建立过程, 是顺序执行的, 故从开始执行第 1 条指令到获得其执行结果所需时间为 3 个基本操作时间之和, 即 $2 + 2 + 1 = 5\ ns$ 。

对于其余 99 条指令均可在该流水线的操作周期时间 ($2\ ns$) 内完成。因此, 这 100 条指令全部执行完毕所需时间为 $5\ ns + 99 \times 2\ ns = 3\ ns + 100 \times 2\ ns = 203\ ns$, 即本题的答案是选项 D。

(5) B。单指令流多数据流 (SIMD) 计算机通常由 1 个控制器、多个执行部件和多个存储器模块组成, 即 SIMD 计算机由 1 个控制部件、多个处理单元同时完成一条指令的执行。所以, 各处理单元必须以同步方式在同一时间内执行相同的指令, 其典型代表是并行处理机 (阵列处理机)。

对于本试题的选项 A, “以同步方式, 在同一时间内执行不同的指令”是针对多指令流多数据流 (MIMD) 来说的, 即 MIMD 是能实现作业、任务、指令等各级全面并行的多机系统, 其典型代表是多处理机。

对于单指令流单数据流 (SISD), 其实质就是传统的顺序执行的单处理器计算机, 其指令部件每次只对一条指令进行译码, 并只对一个操作部件分配数据。

对于多指令流单数据流 (MISD), 其典型结构具有 n 个处理单元, 按 n 条不同指令的要求对同一数据流及其中间结果进行不同的处理。一个处理单元的输出又作为另一个处理单元的输入。这类系统实际上很少见到。

(6) C。当磁头线圈中通入一定方向和大小的脉冲电流时,磁头的导磁体将建立起一定方向和强度的磁场。线圈中电流方向不同,磁介质上被磁化的方向不同,用以代表“1”和“0”。随着电流的变化以及磁介质磁头间的相对运动,就可以把二进制信息序列转化为介质表面的磁化单元序列。由此可见,磁盘驱动器在向盘片的磁性涂层上写入数据时,均以串行方式一位接着一位顺序记录到盘片的磁道上。

(7) D; (8) B。Cache 主要用于解决 CPU 处理器与主存之间存取速度不匹配的问题。由于主存容量为 $2^{12} = 4096$ 块,而每块为 $2^7 = 128$ 个字,因此以字编址的主存总容量为 $4096 \times 128 = 2^{12} \times 2^7 = 2^{19} = 2^9 \times 2^{10} = 512 \text{ K 字}$,故主存地址应为 19 位。

Cache 采用组相联方式映像,其主存地址应分为区号、组号、组内块号、块内地址号等 4 个部分。依题意可知,块内地址号应为 7 位,用以表示 $2^7 = 128$ 个字,每 $2^2 = 4$ 块为一组,则组内块号用 2 位表示。

Cache 容量为 64 块,每 4 块为一组,共分为 $64/4 = 16 = 2^4$ 组,故组号需要用 4 位地址表示。

最后,主存地址 19 位中所剩余的即为区号,应为 $19 - 4 - 2 - 7 = 6$ 位。

(9) B。软件开发包括需求分析、设计、编码、测试和维护等阶段。瀑布模型将软件生命周期划分为制定计划、需求分析、软件设计、程序编写、软件测试和运行维护六个基本活动,并且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。瀑布模型强调文档的作用,并要求每个阶段都要仔细验证。

结构化开发方法的生存周期划分与瀑布模型相对应,因此也是与其最相适应的软件开发方法。

对于本题的选项 A,OMG 描述构件的定义为,构件是一个物理的、可替换的系统组成部分,它包装了实现体且提供了对一组接口的实现方法。构件化方法是以过程建模为先导、以构架为中心、基于构件组装的应用系统开发方法。它可以裁剪为面向对象方法,适合于构件组装模型。构件化方法和面向对象方法鼓励构件组装(复用),面向对象过程沿演化的螺旋迭代,因此该法与瀑布模型软件开发思路不相适应。

对于选项 C,对象是指由数据及其容许的操作所组成的封装体。所谓面向对象,就是基于对象概念,以对象为中心,以类和继承为构造机制,从而认识、理解、刻画客观世界和设计、构建相应的软件系统。面向对象方法是一种把面向对象的思想应用于软件开发过程中,指导开发活动的系统方法,简称 OO (Object Oriented) 方法。

对于选项 D,快速原型模型的第 1 步是建造一个快速原型,实现客户或未来的用户与系统的交互,用户或客户对原型进行评价,进一步细化待开发软件的需求。通过逐步调整原型使其满足客户的要求,开发人员可以确定客户的真正需求是什么;第 2 步则是在第 1 步的基础上开发客户满意的软件产品。显然,快速原型方法可以克服瀑布模型的缺点,减少由于软件需求不明确带来的开发风险,具有显著的效果。

(10) D。需求分析阶段的主要任务是为一个新系统定义业务需求,该阶段的关键是描述一个系统必须做什么(或者一个系统是什么),而不是系统应该如何实现。它通常被划分成五个工作阶段,分别是问题分析、问题评估和方案综合、建模、规约和复审。

具体来说,需求分析阶段需完成以下要求:

- ① 确定软件系统的功能需求和非功能需求;
- ② 分析软件系统的数据要求;
- ③ 导出系统的逻辑模型;
- ④ 修正项目开发计划;
- ⑤ 如有必要,可以开发一个原型系统。

对于本题的选项 D,确定软件系统的运行平台是系统设计阶段的工作任务之一。

(11) C。软件设计通常可分为概要设计和详细设计两个阶段。其中,概要设计的主要任务是:开发软件系统的结构,进行模块划分,确定每个模块的功能、接口以及模块间的调用关系。

体系结构设计的主要目标是开发一个模块化的程序结构,并表示出模块间的控制关系。此外,体系结构设计将程序结构和数据结构相结合,为数据在程序中的流动定义了接口。因此,软件结构设计的主要任务是要确定模块间的组成关系。

选项 A“模块间的操作细节”属于软件物理设计的工作任务之一;选项 D“模块的具体功能”属于软件逻辑设计的工作任务之一。选项 A 及选项 D 均是实现过程中需要考虑的内容。而选项 B“模块间的相似性”则不属于软件结构设计的主要任务。

(12) A。系统测试是将软件系统与硬件、外设和网络等其他因素结合在一起,进行信息系统的各种组装测试和确认测试,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。常见的系统测试主要有恢复测试、强度测试、性能测试、可靠性测试、安装测试和安全性测试。

而对于选项 A,基本路径测试属于白盒测试技术的一种测试方法,该方法允许测试用例设计者导出一个过程设计的逻辑复杂性测度,并使用该测度作为指南来定义执行路径的基本集。从该基本集导出的测试用例保证对程序中的每一条语句至少执行一次。

(13) B。PERT(项目评估与评审技术)图和 Gantt 图(因果分析图,又称鱼骨刺图)是两种常用的项目管理工具。其中,Gantt 图是一种简单的水平条形图,它以一个日历为基准描述项目任务。Gantt 图中横坐标表示时间(如时、天、周、月和年等),纵坐标表示任务,图中的水平线段表示对一个任务的进度安排,线段的起点和终点对应在横坐标上的时间分别表示该任务的开始时间和结束时间,线段的长度表示完成该任务所需的时间。

(14) C。在结构化分析中,数据流图以图形形式描述了系统的逻辑模型,而数据字典是以文字形式描述系统的逻辑模型。数据流图没有对图中各成分进行说明,利用数据字典为数据流图中的每个数据流、数据存储、加工、外部实体以及组成数据流或文件的数据项做出说明。

其中对加工的描述称为“小说明”,或称为“加工逻辑说明”,它描述了输入数据流、输入文件与输出数据流、输出文件之间的逻辑关系。常用的加工逻辑描述方法主要有结构化语言、判定树和判定表。

(15) B。CMM 是对软件组织进化阶段的描述,随着软件组织定义、实施、测量、控制和改进其软件过程,软件组织的能力逐步前进。CMM 将软件过程的成熟度分为 5 个等级,详见表 8-2。

表 8-2 CMM 模型概要

级别	描述	特 点	关键过程域
第 1 级	初始级	软件过程是无序的，几乎没有明确定义的步骤，成功完全依赖个人努力和英雄式的核心任务；企业一般不具备稳定的软件开发与维护的环境，常在遇到问题的时候就放弃原定的计划，而只专注于编程与测试	
第 2 级	可重复级	在这一级别上，建立了基本的项目管理过程来跟踪成本、进度和机能，制定了必要的过程纪律，并基于以往的项目经验来计划与管理新的项目	需求管理、软件配置管理、软件子合同管理等
第 3 级	定义级	管理和工程的软件过程已经文档化、标准化，并综合成整个软件开发组织的标准软件过程。所有的项目都采用根据实际情况修改后得到的标准软件过程来发展和维护软件	组织过程定义、集成软件管理、软件产品工程等
第 4 级	定量管理级	在这一级别制定了软件工程和产品质量的详细度量标准，使用定量分析来不断地改进和管理软件过程。软件过程和产品的质量都被开发组织的成员所理解和控制，因此软件产品具有可预期的高质量	定量的过程管理、软件质量管理
第 5 级	优化级	通过来自过程质量反馈和新观念、新技术的反馈使过程能持续不断地改进。可见整个企业将会把重点放在对过程进行不断的优化上。企业会采取主动找出过程的弱点与优点，以达到预防缺陷的目标	缺陷预防、技术变更管理、过程变更管理

(16) C。结构化设计方法采用结构图(SC)来描述程序的结构。结构图的基本成分由模块、调用和输入/输出数据组成。通常在需求分析阶段，用结构化分析方法产生数据流图。面向数据流的设计能方便地将数据流图(DFD)转换成程序结构图，数据流图中从系统的输入数据到系统的输出数据的一连串连续变换将形成一条信息流。数据流图的信息流大体可分为两种类型：一种是变换流；另一种是事务流。

信息沿着输入通路进入系统，同时将信息的外部形式转换成内部表示，然后通过变换中心处理，再沿着输出通路转换成外部形式化离开系统。具有这种特性的信息流称为变换流。

信息沿着输入通路到达一个事务中心，事务中心根据输入信息的类型在若干个动作序列中选择一个来执行，这种信息流称为事务流。

(17) B。以太网点到点协议(PPoE)是 TCP/IP 协议簇中网络接口层的协议；传输控制协议 TCP 是 TCP/IP 协议簇中传输层的协议；而安全套接层协议(SSL)是工作于 TCP 协议之上的安全协议。

L2F、PPTP、L2TP 是第 2 层的 VPN 协议；IPsec 是第 3 层的 VPN 协议。

(18) B。防火墙的非军事区(DMZ)用于隔离不同网段，是放置公共信息的最佳位置，

所以可在该区域部署本公司对外发布的 Web 服务器,使得外部访问者可以不用通过内网就能直接获得他们所需要的信息。图 8-1 中椭圆虚线圈起的部分是一个非军事区。部分防火墙拓扑结构如图 8-3 所示。

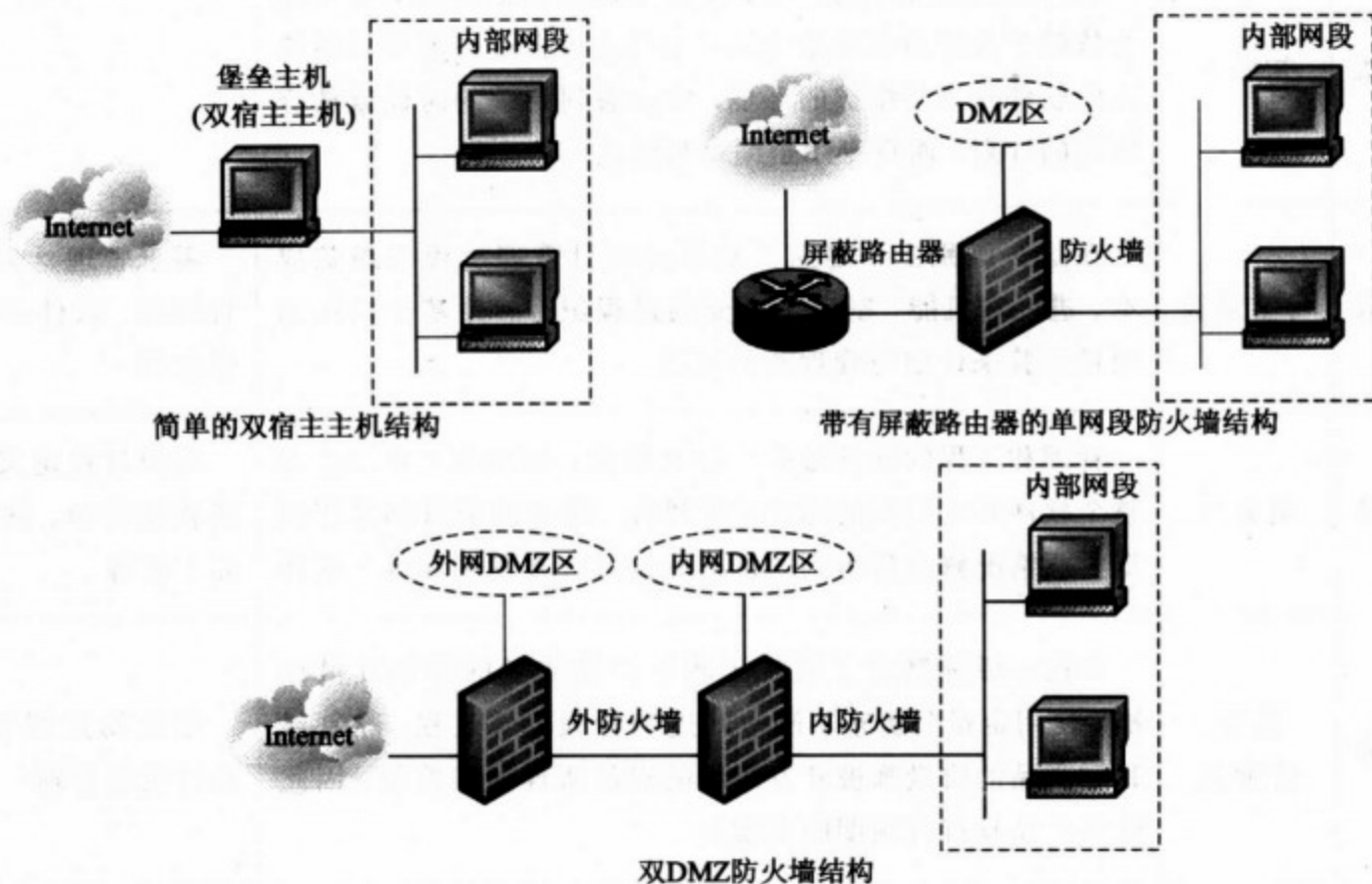


图 8-3 部分防火墙拓扑结构图

(19) B。对于选项 B,抗抵赖性就是防止参与此交易的一方否认曾经发生过此次交易。

对于选项 A,通过身份认证可以确定某一实体的身份,防止一个实体假装成另一个实体。

对于选项 C,认证与授权相结合可以防止他人对数据进行非授权的修改、破坏;

对于选项 D,保护信息的机密性可以防止信息从被监视的通信过程中泄漏出去。

(20) D。知识产权一般都具有法定的保护期限,一旦保护期限届满,权利将自行终止,成为社会公众可以自由使用的知识。至于期限的长短,依各国的法律确定。

对于选项 A,我国发明专利的保护期为 20 年;实用新型专利权和外观设计专利权的期限为 10 年,两者均自专利申请日起计算。

对于选项 B,我国商标权的保护期限自核准注册之日起 10 年,但可以根据其所有人的需要无限期地续展权利期限,在期限届满前 6 个月内申请续展注册,每次续展注册的有效期为 10 年,续展注册的次数不限。如果商标权人逾期不办理续展注册,其商标权也将终止。

对于选项 C,我国公民的作品发表权的保护期为作者终生及其死亡后 50 年。

对于选项 D,商业秘密权受法律保护的期限是不确定的,该秘密一旦为公众所知悉,即成为公众可以自由使用的知识。

(21) D。专利申请具有三个原则,分别是:书面原则、先申请原则和单一性原则。这三个原则的相关描述见表 8-3。

表 8-3 专利申请的三个原则

原 则	描 述
书面原则	是指专利申请人及其代理人在办理各种手续时都应当采用书面形式
先申请原则	是指两人或者两人以上分别就同样的发明创造申请专利的，专利权授给最先申请人
单一性原则	是指一份专利申请文件只能就一项发明创造提出专利申请，即“一申请一发明”原则

甲、乙两人在同一天就同样的发明创造提交了专利申请，这种情况属于同日申请。根据《专利法》第 9 条规定“两个以上的申请人分别就同样的发明创造申请专利的，专利权授予最先申请的人。”以及第 31 条规定“一件发明或者实用新型专利申请应当限于一项发明或者实用新型。属于一个总的发明构思的两项以上的发明或者实用新型，可以作为一件申请提出。一件外观设计专利申请应当限于一种产品所使用的一项外观设计。用于同一类别并且成套出售或者使用的产品的两项以上的外观设计，可以作为一件申请提出。”

对于同日申请的情况，专利局可分别向各申请人通报有关情况，请他们自己去协商解决这一问题，解决的办法一般有两种：一种是两个申请人作为一件申请的共同申请人；另一种是其中一方放弃权利并从另一方得到适当的补偿。如果双方协商不成，则两件申请都不授予专利权。

由以上分析可知，对于选项 D “两件申请都授予专利权”不可能是解决同日申请的办法。

(22) A。根据标准制定的机构和标准适用的范围有所不同，标准可分为国际标准、国家标准、行业标准、企业(机构)标准及项目(课题)标准。根据《中华人民共和国标准化法》的规定，我国标准分为国家标准、行业标准、地方标准和企业标准等四类。这四类标准主要是适用范围不同，不是标准技术水平高低的分级。

对于《计算机软件产品开发文件编制指南》(GB 8567—88)编号中，“GB”表示国家标准的代号，“8567”表示标准发布顺序号，“88”表示标准发布年代号。其中，国家标准 GB 是我国最高标准化机构中华人民共和国国家技术监督局所公布实施的标准，简称为“国标(GB)”。国务院部门，各专业，各省、市、区，各企业，各单位都必须遵守的标准。

根据《中华人民共和国标准化法》关于国家标准、行业标准和地方标准性质的规定，即标准的法律约束性，标准可分为强制性标准、推荐性标准，这两者的区别详见表 8-4。

表 8-4 强制性/推荐性国家标准区别表

原则	代号	描 述	约束性	备 注
强制性国家标准	GB	主要是保障人体健康，人身、财产安全的标准和法律及行政法规规定强制执行的标准	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
推荐性国家标准	GB/T	是指生产、交换、使用等方面，通过经济手段或市场调节而自愿采用的一类标准	不具有强制性，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。推荐性标准一经接受并采用，或各方商定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

推荐性标准是协调一致文件，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，我国制定的《中华人民共和国标准化法》鼓励企业积极采用。推荐性国家标准代号 GB/T 中的“T”是推荐的意思。例如：GB/T 13387—1992《电子材料晶片参考面长度测量方法》系指该标准为推荐性标准。

(23) B；(24) A。程序的局部性理论是指程序执行时，往往会不均匀地访问内存存储器，即有些存储区被频繁访问，有些则少有问津。程序的局部性表现在时间局部性和空间局部性上。

① 时间局部性是指最近被访问的存储单元可能马上又要被访问。产生时间局限性的主要原因是程序中有大量的循环操作。例如程序中的循环体、一些计数变量、累加变量、堆栈等都具有时间局部性特点。

② 空间局部性是指马上被访问的存储单元，其相邻或附近单元也可能马上被访问。例如一段顺序执行的程序、数组的顺序处理等都具有空间局部性特点。

根据程序的局部性理论，Denning 提出了工作集理论。工作集是指进程运行时被频繁访问的页面集合。显然，在进程运行时，如果能保证它的工作集页面都在主存储器内，就会大大减少进程的缺页次数，使进程高效地运行；否则将会因某些工作页面不在内存而出现频繁的页面调入/调出现象，造成系统性能急剧下降，严重时会出现“抖动”现象。

(25) C。在 UNIX 操作系统中，Shell 变量分为参数变量和状态变量两类，见表 8-5。

表 8-5 Shell 定义变量

Shell 参数变量		Shell 状态变量	
\$0	命令名，可用它获得调用该程序的名字	\$?	上一个命令的返回代码，如果命令执行成功，则返回真值，否则返回假值
\$1, \$2, ...	表示 Shell 程序的位置参数	\$ \$	表示当前命令的进程标识数
\$ #	表示位置参数的个数，不包括命令名	\$!	表示 Shell 执行的最近后台进程标识数
\$ *	表示所有位置参量，即相当于 \$1, \$2, \$3	\$ @	表示 Shell 标志位组成的字符串，可以由 Shell 传递，或由 set 命令设置
\$ -	与 \$ * 基本相同，但当用双引号转义时，\$ @ 还是能分解成多个参数，但 \$ * 则合并成一个参数		

注意到试题中“用户键入的命令参数的个数”关键信息，查表 8-5 可知，(25)空缺处的正确答案是 \$ #。

本试题中使用 case 语句实现编程中多选一的控制结构。cat 命令用于查看文件内容，从键盘读取数据，合并文件等。格式为：cat [- b][- A][- E][- T][- n][- s][- v] 文件名。例如，“\$ cat - n file”命令将显示出 file 文件中的行号。

(26) B。快速以太网有多个物理层标准，其物理层规范见表 8-6。

表 8-6 快速以太网物理层规范

标 准	传 输 介 质	特 性 阻 抗	最 大 段 长
100BASE-TX	2 对 5 类 UTP	100 Ω	100 m
	2 对 STP	150 Ω	
100BASE-FX	1 对多模光纤(MMF)	62.5/125 μm	2 km
	1 对单模光纤(SMF)	8/125 μm	40 km
100BASE-T4	4 对 3 类、4 类或 5 类 UTP	100 Ω	100 m
100BASE-T2	2 对 3 类、4 类或 5 类 UTP	100 Ω	

(27) D; (28) C。视频点播 VOD 允许多人观看同一视频节目,它需要用到组播协议。网络会议 Netmeeting 也要用到组播协议,会议小组的成员可能接收同一信息。计算机支持的协同工作(CSCW)需要在组成员之间实时地交换文件、图片或报文,所以也需要组播协议的支持。

网络上的文件传输(FTP)服务通常只允许点对点的信息传输,一般不需要组播技术。

在 IPv4 地址类型中,把 D 类地址作为组播地址,该类地址的头一个字节为 11100000,而其 E 类地址为保留地址。

(29) A。制作双绞线的标准有两个:EIA/TIA 568A 和 EIA/TIA 568B。通常 EIA/TIA 568A 的线序从左至右是:白绿(1)、绿(2)、白橙(3)、蓝(4)、白蓝(5)、橙(6)、白棕(7)、棕(8)。EIA/TIA 568B 的线序从左至右是:白橙(1)、橙(2)、白绿(3)、蓝(4)、白蓝(5)、绿(6)、白棕(7)、棕(8)。两者线序的区别在于(1)和(3),(2)和(6)。

两端都用 EIA/TIA 568B(或 EIA/TIA 568A)标准制作的双绞线称为直通线,用于交换机(或 HUB)与 PC 机网卡的连接,或一台交换机(或 HUB)的以太网端口连接到另一台交换机(或 HUB)的级联端口(UPLINK 口)。

双绞线的一端按照 EIA/TIA 568A 排序,另一端按照 EIA/TIA 568B 排序,这样做出的双绞线就是交叉线。交叉线用于连接两台 PC 机的网卡,或者连接两台交换机(或 HUB)的以太网端口。

(30) D。中间件主要用于三层结构中,充当其结构中的中间层,完成数据安全和完整传输,通过负载均衡来调节系统的工作效率,从而弥补两层结构的不足。所谓三层结构,就是在原有前“两层结构(客户端和服务端)”之间增加一层组件,这层组件包括事务处理逻辑应用服务、数据库查询代理/数据库等。

当前中间件除了消息中间件、数据访问中间件、远程过程调用中间件、对象请求代理中间件和事务处理中间件等 5 大类之外,面向对象的中间件技术已成为中间件平台的主流技术,出现了以 Sun 公司的 EB/J2EE、Microsoft 的 COM+/DNA 和 OMG 的 CORBA/OMA 为代表的三个技术分支。

对于选项 A,Java 远程方法调用(RMI)提供了 Java 程序语言的远程通信功能,这种特性使客户机上运行的程序可以调用远程服务器上的对象,使 Java 编程人员能够在网络环境中分布操作。

对于选项 D,Java Applet 是用 Java 语言编写的一些小应用程序,这些程序是直接嵌入

到页面中,由支持 Java 的浏览器(IE 或 Netscape)解释执行能够产生特殊效果的程序。它可以大大提高 Web 页面的交互能力和动态执行能力。Applet 小应用程序的实现主要依靠 java.applet 包中的 Applet 类。与一般的应用程序不同,Applet 应用程序必须嵌入在 HTML 页面中,才能得到解释执行;同时 Applet 可以从 Web 页面中获得参数,并和 Web 页面进行交互。

(31)~(35) A、A、D、A、B。

参考译文:网络可以用不同的设备互连。在物理层,用中继器或集线器互连,这些设备只是在相同的网络之间传送比特串。在上面的数据链路层,可以使用网桥或交换机,这些设备接收数据帧,检查 MAC 地址,并可以实现少量的协议转换,把数据帧转发到不同的网络中。在网络层,我们使用路由器连接两个网络。如果两个网络的网络层不同,路由器能够对分组格式进行转换。在传输层,我们使用传输网关,它可以在两个传输连接之间建立接口。最后,在应用层,应用网关实现消息语法之间的翻译。例如,在 Internet 邮件和 X.400 邮件之间的网关可以对邮件报文进行语法分析,对报文的各个报头字段做出改变。

(36)~(40) C、A、B、A、B。

参考译文:需求定义阶段的作用就是为软件产品产生一个清晰的、完整的、一致的、可测试的技术需求规格说明。

在需求定义阶段,需求定义团队通过迭代过程把有关系统需求的宽泛的陈述扩展成系统必须实现的、各个功能的、完整而详细的规格说明,以及系统应该满足的各种准则。这个过程的起点通常是由用户提供的描述项目或问题的一组高级需求。

在任何情况下,需求定义团队应该阐明与系统有关的整体概念,并且设定一种能够显示系统如何运作的情景,发布有关系统和运行的概念文档,并进行系统概念的评审(SCR)。

在 SCR 之后,团队应该根据高级需求和系统运行概念导出详细的系统需求。通过结构化或面向对象的分析技术,团队可以规范系统功能,说明满足每一细节需求的算法。

8.2 2005 年上半年真题链接

8.2.1 上午试题

• 在计算机中,最适合进行数字加减运算的数字编码是(1),最适合表示浮点数阶码的数字编码是(2)。

- (1) A. 原码 B. 反码 C. 补码 D. 移码
(2) A. 原码 B. 反码 C. 补码 D. 移码

• 如果主存容量为 16 MB,且按字节编址,则表示该主存地址至少应需要(3)位。

- (3) A. 16 B. 20 C. 24 D. 32

• 操作数所处的位置,可以决定指令的寻址方式。操作数包含在指令中,寻址方式为(4);操作数在寄存器中,寻址方式为(5);操作数的地址在寄存器中,寻址方式为(6)。

- (4) A. 立即寻址 B. 直接寻址
 C. 寄存器寻址 D. 寄存器间接寻址
(5) A. 立即寻址 B. 相对寻址

- C. 寄存器寻址
D. 寄存器间接寻址
- (6) A. 相对寻址
B. 直接寻址
- C. 寄存器寻址
D. 寄存器间接寻址

• 三个可靠度 R 均为 0.8 的部件串联构成一个系统, 如图 8-4 所示, 则该系统的可靠度为(7)。

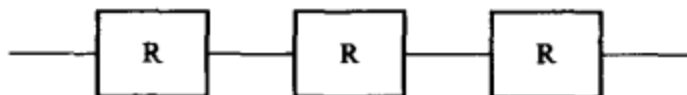


图 8-4 某系统组成结构图

- (7) A. 0.240 B. 0.512 C. 0.800 D. 0.992
- 在计算机系统中, 构成虚拟存储器(8)。
- (8) A. 只需要一定的硬件资源便可实现
B. 只需要一定的软件即可实现
C. 既需要软件也需要硬件方可实现
D. 既不需要软件也不需要硬件
- 某公司使用包过滤防火墙控制进出公司局域网的数据, 在不考虑使用代理服务器的情况下, 下面描述错误的是“该防火墙能够(9)”。
- (9) A. 使公司员工只能访问 Internet 上与其有业务联系的公司的 IP 地址
B. 仅允许 HTTP 协议通过
C. 使员工不能直接访问 FTP 服务器端口号为 21 的 FTP 地址
D. 仅允许公司中具有某些特定 IP 地址的计算机可以访问外部网络
- 两个公司希望通过 Internet 进行安全通信, 保证从信息源到目的地之间的数据传输以密文形式出现, 而且公司不希望由于在传输节点使用特殊的安全单元增加开支, 最合适的加密方式是(10), 使用的会话密钥算法应该是(11)。
- (10) A. 链路加密 B. 节点加密
C. 端-端加密 D. 混合加密
- (11) A. RSA B. RC-5
C. MD5 D. ECC
- 我国著作权法中, (12)系指同一概念。
- (12) A. 出版权与版权 B. 著作权与版权
C. 作者权与专有权 D. 发行权与版权
- 由我国信息产业部批准发布, 在信息产业部门范围内统一使用的标准, 称为(13)。
- (13) A. 地方标准 B. 部门标准
C. 行业标准 D. 企业标准
- 某软件设计师自行将他人使用 C 程序语言开发的控制程序转换为机器语言形式的控制程序, 并固化在芯片中, 该软件设计师的行为(14)。
- (14) A. 不构成侵权, 因为新的控制程序与原控制程序使用的程序设计语言不同
B. 不构成侵权, 因为对原控制程序进行了转换与固化, 其使用和表现形式不同
C. 不构成侵权, 将一种程序语言编写的源程序转换为另一种程序语言形式, 属

于一种“翻译”行为

D. 构成侵权，因为他不享有原软件作品的著作权

• 数据存储在磁盘上的排列方式会影响 I/O 服务的总时间。假设每磁道划分成 10 个物理块，每块存放 1 个逻辑记录。逻辑记录 R1, R2, …, R10 存放在同一个磁道上，记录的安排顺序如表 8-7 所示。

表 8-7 逻辑记录安排顺序表

物理块	1	2	3	4	5	6	7	8	9	10
逻辑记录	R1	R2	R3	R4	R5	R6	R7	R8	R9	R10

假定磁盘的旋转速度为 20 ms/周，磁头当前处在 R1 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 4 ms，则处理这 10 个记录的最长时间为(15)；若对信息存储进行优化分布后，处理 10 个记录的最少时间为(16)。

- (15) A. 180 ms B. 200 ms C. 204 ms D. 220 ms
- (16) A. 40 ms B. 60 ms C. 100 ms D. 160 ms

• 页式存储系统的逻辑地址是由页号和页内地址两部分组成的。假定页面的大小为 4 KB，地址变换过程如图 8-5 所示，该图中逻辑地址用十进制表示。

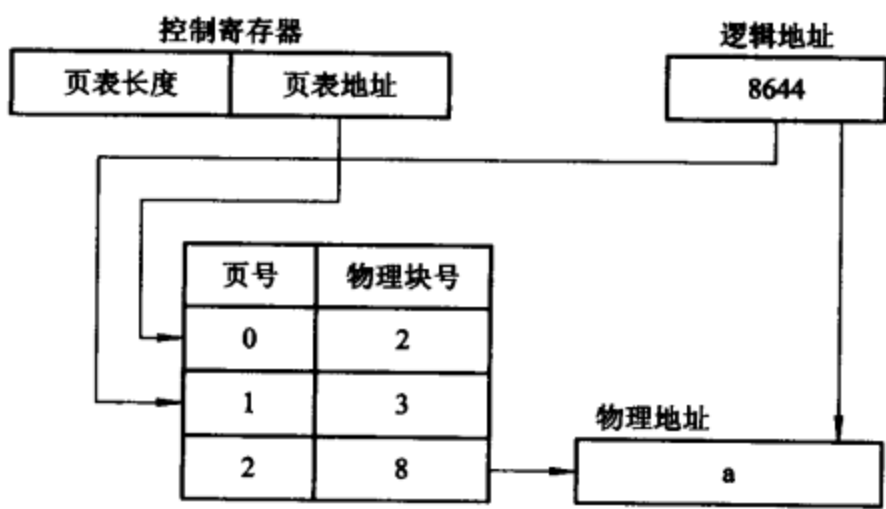


图 8-5 页式存储系统地址变换示意图

图 8-5 中有效地址经过变换后，十进制物理地址 a 应为(17)。

- (17) A. 33220 B. 8644 C. 4548 D. 2500

• 下列叙述中，与提高软件可移植性相关的是(18)。

- (18) A. 选择时间效率高的算法
- B. 尽可能减少注释
- C. 选择空间效率高的算法
- D. 尽量用高级语言编写系统中对效率要求不高的部分

• 在系统转换的过程中，旧系统和新系统并行工作一段时间，再由新系统代替旧系统的策略称为(19)；在新系统全部正式运行前，一部分一部分地代替旧系统的策略称为(20)。

- (19) A. 直接转换 B. 位置转换
 C. 分段转换 D. 并行转换
(20) A. 直接转换 B. 位置转换
 C. 分段转换 D. 并行转换

• 一个局域网中某台主机的 IP 地址为 176.68.160.12, 使用 22 位作为网络地址, 那么该局域网的子网掩码为(21), 最多可以连接的主机数为(22)。^①

- (21) A. 255.255.255.0 B. 255.255.248.0
 C. 255.255.252.0 D. 255.255.0.0
(22) A. 254 B. 512 C. 1022 D. 1024

• 以下选项中, 可以用于 Internet 信息服务器远程管理的是(23)。

- (23) A. Telnet B. RAS C. FTP D. SMTP

• 在 TCP/IP 网络中, 为各种公共服务保留的端口号范围是(24)。

- (24) A. 1~255 B. 1~1023
 C. 1~1024 D. 1~65535

• 在以下网络应用中, 要求带宽最高的应用是(25)。

- (25) A. 可视电话 B. 数字电视
 C. 拨号上网 D. 收发邮件

• DOM is a platform-and-language-(26) API that allows programs and scripts to dynamically access and update the content, structure and style of WWW documents (currently, definitions for HTML and XML documents are part of the specification). The document can be further processed and the results of that processing can be incorporated back into the presented(27). DOM is a (28)-based API to documents, which requires the whole document to be represented in (29) while processing it. A simpler alternative to DOM is the event - based SAX, which can be used to process very large (30) documents that do not fit into the memory available for processing.

- (26) A. specific B. neutral C. contained D. related
(27) A. text B. image C. page D. graphic
(28) A. table B. tree C. control D. event
(29) A. document B. processor C. disc D. memory
(30) A. XML B. HTML C. script D. web

• Melissa and Love Letter made use of the trust that exists between friends or colleagues. Imagine receiving an (31) from a friend who asks you to open it. This is what happens with Melissa and several other similar E-mail (32). Upon running, such worms usually proceed to send themselves out to E-mail addresses from the victim's address book, previous E-mails, web pages (33).

As administrators seek to block dangerous E-mail attachments through the recognition of well-known (34), virus writers use other extensions to circumvent such protection.

^① 注: 试题编号(21)~(35)在原试题中的编号为(61)~(75)。

Executable (.exe) files are renamed to .bat and .cmd plus a whole list of other extensions and will still run and successfully infect target users.

Frequently, hackers try to penetrate networks by sending an attachment that looks like a flash movie, which, while displaying some cute animation, simultaneously runs commands in the background to steal your passwords and give the (35) access to your network.

- (31) A. attachment B. packet C. datagram D. message
 (32) A. virtual B. virus C. worms D. bacteria
 (33) A. memory B. caches C. ports D. registers
 (34) A. names B. cookies C. software D. extensions
 (35) A. cracker B. user C. customer D. client

8.2.2 参考答案

表 8-8 给出本份真题问题(1)~(35)的参考答案,供读者练习时参考,以便查缺补漏。读者可按每空 1 分的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识的掌握程度。

表 8-8 参考答案及评分标准

问题号	参考答案	问题号	参考答案
(1)~(5)	C、D、C、A、C	(21)~(25)	C、C、A、B、B
(6)~(10)	D、B、C、B、C	(26)~(30)	B、C、B、D、A
(11)~(15)	B、B、C、D、C	(31)~(35)	A、C、B、D、A
(16)~(20)	B、A、D、D、C		

8.2.3 要点解析

(1) C; (2) D。本题考查的知识点是机内数据的表示形式。进行原码加减运算时,操作数与运算结果均用原码表示。由于原码做加减运算时符号位要单独处理,使得运算较复杂,因此在计算机中很少被采用。

为了简化运算方法,常采用补码表示法,以便符号位也能作为数值的一部分参与运算。补码加法的运算法则是:和的补码等于补码求和。补码减法的运算法则是:差的补码等于被减数的补码加上减数取负后的补码。负数补码表示的实质是将负数映射到正数域,所以可将减法运算转化为加法运算。在补码加减运算中,符号位和数值位一样参加运算,无需做特殊处理。因此,多数计算机都采用补码加减运算法。

移码是机器数的另一种表示方法,又称增码,最适合表示浮点数的阶码。移码的符号位用 1 表示正号,用 0 表示负号,求法是把其补码的符号位直接取反即可。可见,正数的移码大于负数的移码,利用这一特点,移码被广泛用来表示浮点数阶码的数字编码,从而只比较阶码的大小就可以实现真值大小的比较。

由以上分析的结果可知,本题答案应该是(1)C,(2)D。

(3) C。用二进制编码表示地址, $1\text{ KB}=2^{10}\text{ B}$, $1\text{ MB}=2^{20}\text{ B}$ 。因为 $2^4=16$, 所以 16 MB 地址至少应需要 $2^4 \times 2^{20} = 2^{24}$ 位。

(4) A; (5) C; (6) D。本试题选项中几种寻址方式的相关描述及其区别见表 8-9。

表 8-9 几种寻址方式的相关描述及其区别

寻址方式	描 述	示 例
立即寻址	也称为立即数寻址, 操作数在指令中直接给出	ADD R0, R0 #0x3F /* R0←R0+0x3F */
寄存器寻址	操作数就是在内部寄存器中, 寄存器名在指令中给出	ADD R0, R1, R2 /* R0←R1+R2 */
寄存器间接寻址	以寄存器中的值作为操作数的地址, 而操作数本身存放在存储器中	ADD R0, R1, [R2] /* R0←R1+[R2] */
直接寻址	操作数总是在存储器中, 存储单元的有效地址由指令直接指出	ADD R0, R1, [0x3F] /* R0←R1+[0x3F] */
相对寻址	以程序计数器 PC 的当前值为基地址, 指令中的地址标号作为偏移量, 将两者相加之后得到操作数的有效地址	

(7) B。本试题中 3 个部件串联构成一个系统, 3 个部件中任何一个失效就足以使系统失效。该串联系统的可靠度为 $R_s = R \times R \times R = 0.8 \times 0.8 \times 0.8 = 0.512$ 。

(8) C。虚拟存储器是扩大内存容量、解决容量与速度矛盾的有效措施。用户编程时无需考虑计算机的实际内存容量, 可以写出比实际配置的内存容量大得多的用户程序。

虚拟存储器是一种通过硬件和软件的综合来扩大用户可用存储空间的技术。系统软件操作系统通过一定的调度策略, 把外存当内存来使用; 程序预先放在外存, 在操作系统的管理和调度下, 按某种算法依次调入内存被 CPU 执行。可见, 在计算机系统中, 构成虚拟存储器既需要硬件也需要管理软件方可实现。

(9) B。IP 包过滤防火墙工作在 TCP/IP 协议簇的网络层, 通常有一个包检查模块(通常称为包过滤器)。对于数据包的过滤可以根据数据包中的 IP 地址、端口号等信息来控制站点与站点、站点与网络、网络与网络之间的相互访问, 但无法控制传输中的用户数据内容。因为用户数据属于应用层数据。无论是源 IP 地址还是目的 IP 地址, 都是网络层的 IP 地址, 都在包过滤防火墙的控制范围内, 因此, 通过配置目的 IP 和源 IP, 就可以实现选项 A 和选项 D 的访问控制规则。

虽然 FTP 是应用层协议,但是默认情况下,FTP 协议开放的是传输控制协议(TCP)的 21 端口号,通过包过滤防火墙限制 TCP 端口号,可以实现选项 C 的访问控制规则。

HTTP 协议是超文本传输协议,它是应用层协议,包过滤防火墙无法实现对应用层协议的限制,所以无法实现选项 B 的访问控制规则。

(10) C; (11) B。通常信息的传输过程中的加密方式有链路加密、节点加密和端-端加密三种方式,见表 8-10。

表 8-10 信息的传输加密方式

加密方式	描 述
链路加密	只对两个节点之间(不含信源和信宿两个端点)的通信信道线路上所传输的信息进行加密保护,但是在传输过程中经过每个节点时,节点中的数据是明文
节点加密	每个节点里装有加解密保护装置,用于完成一个密钥向另一个密钥的转换。节点中虽然不会出现明文,但是需要在经过的每个节点加装保护装置,这不仅不方便使用,而且会增加开支
端-端加密	为系统提供从信源到信宿传送数据的加密保护,不需要在通信节点上增加额外的安全单元,而且能够保证数据自始至终以密文形式出现,即使在节点中也是密文

RC-5 算法是对称密码算法,加解密都使用相同的密钥,加密效率高,适合于加密大量的数据。

RSA 和 ECC 是非对称密码算法,加解密过程使用不同的密钥(即公钥和私钥),它们对计算资源的消耗较大,适合于加密非常少量的数据,例如加密会话密钥。

MD5 算法主要用于生成数字摘要。

(12) B。我国著作权法第 56 条中指出:“本法所称的著作权即版权”,即著作权与版权系指同一概念。

(13) C。根据《中华人民共和国标准化法》的规定,我国标准分为国家标准、行业标准、地方标准和企业标准等四类。这四类标准主要是适用范围不同,不是标准技术水平高低的分级。

由国务院有关行政主管部门制定并报国务院标准化行政主管部门备案的标准,称为行业标准。我国信息产业部属我国行政主管部门,其批准发布标准在信息行业范围内为行业统一的标准。

(14) D。计算机软件著作权的客体是指著作权法保护的计算机软件著作权的范围(受保护的客体)。著作权法保护的计算机软件著作权是指计算机程序(源程序和目标程序)及其有关文档(程序设计说明书、流程图、用户手册等)。

本试题中该软件设计师自行使用他人用 C 程序语言所开发的软件的行为属于侵权行为。因为他未经软件著作权人的许可。

(15) C; (16) B。本题的解答思路如下:

① 依题意“每磁道划分成10个物理块,每块存放1个逻辑记录”和“磁盘的旋转速度为20 ms/周”可得,系统读取每一个逻辑记录的时间 $t_1 = 20 \text{ ms} / 10 = 2 \text{ ms}$ 。

② 如果逻辑记录 R1~R10 按表 8-7 顺序存放在同一个磁道,依题意“系统使用单缓冲区,每个记录处理时间为 4 ms”,那么当系统读出并处理完逻辑记录 R1,所花费的时间是 $t_2 = 2 \text{ ms} + 4 \text{ ms} = 6 \text{ ms}$ 。

③ 由于处理完逻辑记录 R1 之后,此时磁头已转到记录 R4 的开始处,因此为了读出逻辑记录 R2,磁盘必须继续转至逻辑记录 R2 的开始处,这一过程磁头将经过 8 个逻辑记录,所花费的时间为 $t_3 = 16 \text{ ms}$ 。

由此可见,系统从处理完逻辑记录 R1 到处理完逻辑记录 R2 所花费的时间是 $t_4 = 16 \text{ ms} + 2 \text{ ms} + 4 \text{ ms} = 22 \text{ ms}$ 。

④ 对于逻辑记录 R3~R10 处理时间的分析过程类似逻辑记录 R2 的分析过程,因此,处理按表 8-7 顺序存放的 R1~R10 这 10 个记录的总时间 $t_5 = t_2 + t_4 \times 9 = 6 \text{ ms} + 22 \text{ ms} \times 9 = 204 \text{ ms}$ 。

⑤ 若要求对信息存储进行优化分布使处理 10 个逻辑记录的时间最少,那么就必须满足当读出前一个逻辑记录(例如 R1)并处理结束后,磁头刚好转至下一个记录(例如 R2)的开始处。

⑥ 由读取一个逻辑记录的时间为 2 ms,处理一个逻辑记录的时间为 4 ms 可推理出,如逻辑记录 R1~R10 按表 8-11 进行分布优化,则处理 10 个记录的时间最少。

表 8-11 优化分布后逻辑记录安排顺序

物理块	1	2	3	4	5	6	7	8	9	10
逻辑记录	R1	R8	R5	R2	R9	R6	R3	R10	R7	R4

⑦ 优化分布后处理 10 个记录的总时间为 $t_6 = 6 \text{ ms} \times 10 = 60 \text{ ms}$ 。

(17) A。本题的解答思路如下:

① 在页式存储管理方案中,有效地址除页面的大小,所得结果取整为页号,取余为页内地址。

② 依题可知,该页式存储系统的页面的大小为 $4 \text{ KB} = 4 \times 2^{10} = 4096 \text{ B}$,有效逻辑地址为 8644。

$8644 / 4096 = 2 \cdots 452$,其中商 2 为页号,余数 452 为页内地址。

③ 查图 8-5 的页表,得页号为 2 所对应的物理块号为 8,因此 a 的物理地址为 $8 \times 4096 + 452 = 33220$ 。

(18) D。软件可移植性是指与软件从某一环境移植到另一环境的能力有关的一组属性。

通常高级语言具有较好的可移植性,因此编辑时应尽量用高级语言编写系统中对效率要求不高的程序部分。采用减少注释、选择时间/空间效率高的算法都不能提高软件的可移植性。

(19) D; (20) C。新系统试运行成功之后,就可以在新系统和旧系统之间互相转换。新旧系统之间的转换方式通常有直接转换、并行转换和分段转换,见表 8-12。

表 8-12 系统之间的转换方式

转换方式	描 述	备 注
直接转换	是指在确定新系统运行无误时，立刻启用新系统，终止旧系统运行	这种转换方式对人员、设备费用很节省。这种方式一般适用于一些处理过程不太复杂，数据不太重要的场合
并行转换	是指新旧系统并行工作一段时间，经过一段时间的考验以后，新系统正式替代旧系统。对于较复杂的大型系统，它提供了一个与旧系统运行结果进行比较的机会，可以对新旧两个系统的时间要求、出错次数和工作效率给以公正的评价	这是一种经常使用的转换方式。由于与旧系统并行工作，因此消除了尚未认识新系统之前的紧张和不安。其特点是安全、可靠，但费用和工作量都很大，因为在相当长时间内系统要两套班子并行工作
分段转换	又称逐步转换、向导转换、试点过渡法等。这种转换方式实际上是直接转换与并行转换方式的结合。在新系统全部正式运行前，一部分一部分地代替旧系统。那些在转换过程中还没有正式运行的部分，可以在一个模拟环境中继续试运行	这种转换方式既保证了可靠性，又不至于费用太大。但是这种分段转换要求子系统之间有一定的独立性，对系统的设计和实现都有一定的要求，否则就无法实现这种分段转换的设想

(21) C; (22) C。子网掩码的作用是将某个 IP 地址划分成网络地址和主机地址两部分。与 IP 地址相同，子网掩码的长度也是 32 位，左边是网络位，用二进制数字“1”表示；右边是主机位，用二进制数字“0”表示。在一个局域网中，如果某台主机的 IP 地址使用 22 位作为网络地址，则子网掩码的二进制表示为：11111111 11111111 11111100 00000000，即十进制的 255.255.252.0。

由于局域网中某台主机的 IP 地址为 176.68.160.12，如果使用 22 位作为网络地址，那么主机地址用 $32-22=10$ 位来表示，则在该局域网中最多可以连接的主机数为： $2^{10}-2=1024-2=1022$ 。

其中“-2”的意义是：在该局域网中最多可以连接的主机数减去 176.68.160.0(该网段的网络地址)和 176.68.163.255(该网段的广播地址)这两个 IP 地址。

(23) A。对于选项 A，Telnet 是基于客户机/服务器模式的服务系统，它由客户软件、服务器软件以及 Telnet 通信协议等三部分组成。远程登录服务是在 Telnet 协议的支持下，将用户计算机与远程主机连接起来，在远程计算机上运行程序，将相应的屏幕显示传送到本地机器，并将本地的输入送给远程计算机，因此 Telnet 远程登录可以用于 Internet 信息服务器远程管理。

对于选项 B，RAS(Remote Access Service)用于远程访问服务，例如使用拨号上网时，在远端的服务器上需要启动远程访问服务 RAS。

对于选项 C, FTP 是文件传输协议, 用来通过网络从一台计算机向另一台计算机传送文件, 它是互联网上继 WWW 服务之后的另一项主要服务, 用户可以通过 FTP 客户程序连接 FTP 服务器, 然后利用 FTP 协议进行文件的下载或上传。

对于选项 D, SMTP 是简单邮件管理协议, 是一种通过计算机网络与其他用户进行联系的快速、简便、高效、廉价的现代化通信手段, 是一种利用网络交换信息的非交互式服务。在 TCP/IP 网络上的大多数邮件管理程序使用 SMTP 协议来发信, 且采用 POP3 协议来保管用户未能及时取走的邮件。

(24) B。在 TCP/IP 协议簇中, 上层协议实体与下层协议实体之间的逻辑接口叫做服务访问点(SAP)。在因特网中, 网卡接口是物理层的服务访问点; IP 地址是网络层的服务访问点; 端口号是传输层的服务访问点; 应用层的服务访问点是用户界面。

每一个应用服务均对应有一个统一的端口号, 对于各种公共服务保留的端口号范围为 1~1023。例如 SMTP 服务所用的 TCP 端口号是 25; POP 服务所用的 TCP 端口号是 110; WWW 服务使用的 TCP 端口为 80 等。如果用户改变了服务器上各种服务保留的端口号, 那么需要注意的是在客户端使用相应服务时也要改为同一端口号。

(25) B。在拨号上网和收发邮件应用中, 带宽要求并不高, 只是带宽高时拨号上网和收发邮件的速度提高了。对于可视电话和数字电视均要求带宽越高越好, 而对于数字电视尤其如此, 因为每一帧图像, 如果每一点像素使用 24 位作为它的颜色值, 设数字电视的分辨率为 1920×1080 , 为使图像不抖动, 每秒钟至少需要 25 帧, 则每秒需要传送的字节数为: $1920 \times 1080 \times 24 \times 25 / 8 = 155\,520\,000\text{ B} = 155\text{ MB}$ 。

由此可见, 在本试题中的 4 个选项中, 数字电视对网络带宽的要求最高。

(26)~(30) B、C、B、D、A。

参考译文: DOM 是一种与平台和语言无关的应用程序接口(API), 它可以动态地访问程序和脚本, 更新其内容、结构和 WWW 文档的风格(目前, HTML 和 XML 文档是通过说明部分定义的)。文档可以进一步被处理, 处理的结果可以加入到当前的页面。DOM 是一种基于树的 MI 文档, 它要求在处理过程中整个文档都表示在存储器中。另外一种简单的 API 是基于事件的 SAX, 它可以用于处理很大的 XML 文档, 由于该文档较大, 因此不适合全部放在存储器中处理。

(31)~(35) A、C、B、D、A。

参考译文: Melissa 和 Love Letter 利用了朋友和同事之间的信任。设想你收到了一个附件, 朋友要求你打开它。这就是 Melissa 和其他类似的电子邮件蠕虫危害的方式。一旦被运行, 这种蠕虫常常就从受害人的地址簿、以前的电子邮件、Web 页面缓存区等地方发现邮件地址, 并且把它们发送出去。

在管理人员通过识别众所周知的扩展名企图阻挡危险的 E-mail 附件时, 病毒的作者们则使用了其他的扩展名来破坏这种保护。可执行文件(.exe)被重新命名为 .bat 和 .cmd, 再加上其他的扩展名, 而且仍然可以运行, 成功地危害目标用户。

黑客们经常试图穿透网络发送一种像是 flash 影像的附件, 当这种 flash 演示引人入胜的动画时, 同时就在后台运行命令来偷窃你的口令, 给了那些解密高手访问你的网络的机会。

8.3 2005 年下半年真题链接

8.3.1 上午试题

- 阵列处理机属于(1)计算机。

(1) A. SISD B. SIMD C. MISD D. MIMD

- 采用(2)不能将多个处理机互联构成多处理机系统。

(2) A. STD 总线 B. 交叉开关
C. PCI 总线 D. Centronic 总线

- 某计算机系统的可靠性结构为如图 8-6 所示的双重串并联结构,若所构成系统的每个部件的可靠度为 0.9,即 $R=0.9$,则系统的可靠度为(3)。

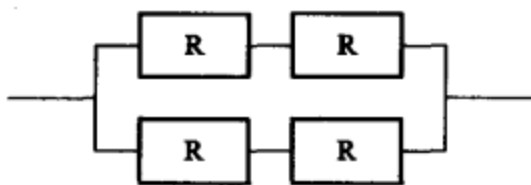


图 8-6 某计算机系统的组成结构图

(3) A. 0.9997 B. 0.9276 C. 0.9739 D. 0.6561

- 若每一条指令都可以分解为取指、分析和执行三步。已知取指时间 $t_{\text{取指}}=5\Delta t$, 分析时间 $t_{\text{分析}}=2\Delta t$, 执行时间 $t_{\text{执行}}=5\Delta t$ 。如果按顺序方式从头到尾执行完 500 条指令需(4) Δt 。如果按照[执行] k 、[分析] $k+1$ 、[取指] $k+2$ 重叠的流水线方式执行指令,从头到尾执行完 500 条指令需(5) Δt 。

(4) A. 5590 B. 5595 C. 6000 D. 6007

(5) A. 2492 B. 2500 C. 2510 D. 2515

- 在开发一个系统时,如果用户对系统的目标不很清楚,难以定义需求,这时最好使用(6)。

(6) A. 原型法 B. 瀑布模型 C. V-模型 D. 螺旋模型

- 应该在(7)阶段制定系统测试计划。

(7) A. 需求分析 B. 概要设计 C. 详细设计 D. 系统测试

- 已经发布实施的标准(包括已确认或修改补充的标准)经过实施一定时期后,对其内容再次审查,以确保其有效性、先进性和适用性,其周期一般不超过(8)年。

(8) A. 1 B. 3 C. 5 D. 7

- (9)不需要登记或标注版权标记就能得到保护。

(9) A. 专利权 B. 商标权 C. 著作权 D. 财产权

- 在图 8-7 所示的树型文件中,方框表示目录,圆圈表示文件,“/”表示路径的分隔符,“/”于路径之首表示根目录。在图 8-7 中,(10)。

假设当前目录是 D1,进程 A 以如下两种方式打开文件 f1。

① `fd1=open("(11)/f1", o_RDONLY);`

② `fd1=open("/D1/W1/f1", o_RDONLY);`

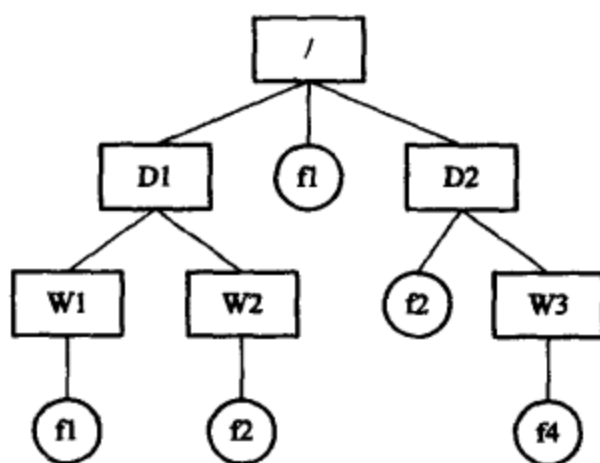


图 8-7 某文件系统结构图

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 的文件系统(12)。

- (10) A. 子目录 W2 中文件 f2 和子目录 D2 中文件 f2 是完全相同的
 B. 子目录 W2 中文件 f2 和子目录 D2 中文件 f2 是不相同的
 C. 子目录 W2 中文件 f2 和子目录 D2 中文件 f2 可能相同也可能不相同
 D. 树型文件系统中不允许出现相同名字的文件

- (11) A. /D1/W1 B. D1/W1 C. W1 D. f1

- (12) A. 可以直接访问根目录下的文件 f1
 B. 可用从当前路径开始查找需要访问的文件 f1
 C. 只需要访问一次磁盘，就可以读取文件 f1，而方式 2 需要两次
 D. 只需要访问一次磁盘，就可以读取文件 f1，而方式 2 需要三次

• 以下内容中，(13)应写入操作手册。

- (13) A. 描述系统对各种输入数据的处理方法
 B. 说明系统升级时厂商提供的服务
 C. 描述系统处理过程的各个界面
 D. 说明系统各部分之间的接口关系

• 代码走查(code walkthrough)和代码审查(code inspection)是两种不同的代码评审方法，这两种方法的主要区别是(14)。

- (14) A. 在代码审查中由编写代码的程序员来组织讨论，而在代码走查中由高级管理人员来领导评审小组的活动
 B. 在代码审查中只检查代码是否有错误，而在代码走查中还要检查程序与设计文档的一致性
 C. 在代码走查中只检查程序的正确性，而在代码审查中还要评审程序员的编程能力和工作业绩
 D. 代码审查是一种正式的评审活动，而代码走查的讨论过程是非正式的

• 在软件项目管理中可以使用各种图形工具来辅助决策，下面对 Gannt 图的描述不正确的是(15)。

- (15) A. Gannt 图表现各个活动的顺序和它们之间的因果关系
 B. Gannt 图表现哪些活动可以并行进行
 C. Gannt 图表现了各个活动的起始时间

D. Gantt 图表现了各个活动完成的进度

- 关于维护软件所需的成本, 以下叙述正确的是(16)。

(16) A. 纠正外部和内部设计错误比纠正源代码错误需要更大的成本

B. 与需求定义相比, 源代码的文字量大得多, 所以源代码的维护成本更高

C. 用户文档需要经常更新, 其维护成本超过了纠正设计错误的成本

D. 需求定义的错误会在设计时被发现并纠正, 因此需求定义纠错的成本小于源代码纠错的成本

- 一个故障已经被发现, 而且也被排除了, 为了检查修改是否引起了其他故障, 这时应该进行(17)。

(17) A. 程序走查

B. 退化测试

C. 软件评审

D. 接收测试

- 假设有一个局域网, 管理站每 15 分钟轮询被管理设备一次, 一次查询访问需要的时间是 200 ms, 则管理站最多可以支持(18)个网络设备。

(18) A. 400

B. 4000

C. 4500

D. 5000

- 使用 RAID 作为网络存储设备有许多好处, 以下关于 RAID 的叙述中不正确的是(19)。

(19) A. RAID 使用多块廉价磁盘阵列构成

B. RAID 采用交叉存取技术, 提高了访问速度

C. RAID0 使用磁盘镜像技术, 提高了可靠性

D. RAID3 利用一个奇偶校验盘完成容错功能, 减少了冗余磁盘数量

- 为保证用户在网络上边下载边观看视频信息, 需要采用(20)技术。

(20) A. 流媒体

B. 数据库

C. 数据采集

D. 超链接

- 通过代理服务器使内部局域网中的客户机访问 Internet 时, (21)不属于代理服务器的功能。

(21) A. 共享 IP 地址

B. 信息缓存

C. 信息转发

D. 信息加密

- 下列(22)设备可以隔离 ARP 广播帧。

(22) A. 路由器

B. 网桥

C. 以太网交换机

D. 集线器

- 在 Windows 系统中, (23)不是网络服务组件。

(23) A. RAS

B. HTTP

C. IIS

D. DNS

- 在 OSI 参考模型中, 数据链路层处理的数据单位是(24)。

(24) A. 比特

B. 帧

C. 分组

D. 报文

- 在 OGSA 标准中定义了(25)的概念, 它提供一组遵守特定的约定并定义明确的接口, 是实体之间产生、管理和交换信息的机制。

(25) A. Object

B. Grid Service

C. Web Service

D. XML

- MIDI enables people to use (26) computers and electronic musical instruments. There are actually three components to MIDI, the communications "(27)", the Hardware Interface and a distribution (28) called "Standard MIDI Files". In the context of the

WWW, the most interesting component is the (29)Format. In principle, MIDI files contain sequences of MIDI Protocol messages. However, when MIDI Protocol (30)are stored in MIDI files, the events are also time-stamped for playback in the proper sequence. Music delivered by MIDI files is the most common use of MIDI today.

- (26) A. personal B. electronic C. multimedia D. network
 (27) A. device B. protocol C. network D. controller
 (28) A. format B. text C. wave D. center
 (29) A. Video B. Faxmail C. Graphic D. Audio
 (30) A. messages B. packets C. frame D. information

• Certificates are (31) documents attesting to the (32) of a public key to an individual or other entity. They allow verification of the claim that a given public key does in fact belong to a given individual. Certificates help prevent someone from using a phony key to (33)someone else. In their simplest form, Certificates contain a public key and a name. As commonly used, a certificate also contains an (34)date, the name of the CA that issued the certificate, a serial number, and perhaps other information. Most importantly, it contains the digital (35) of the certificate issuer. The most widely accepted format for certificates is X. 509, thus, Certificates can be read or written by any application complying with X. 509.

- (31) A. text B. data C. digital D. structured
 (32) A. connecting B. binding C. composing D. conducting
 (33) A. impersonate B. personate C. damage D. control
 (34) A. communication B. computation
 C. expectation D. expiration
 (35) A. signature B. mark C. stamp D. hypertext

8.3.2 参考答案

表 8-13 给出本份真题问题(1)~(35)的参考答案,供读者练习时参考,以便查缺补漏。读者可按每空 1 分的评分标准得出自己的测试分数,从而大致评估自己对这一方面知识点的掌握程度。

表 8-13 参考答案及评分标准

问题号	参考答案	问题号	参考答案
(1)~(5)	B、D、C、C、C	(21)~(25)	D、A、B、B、B
(6)~(10)	A、A、C、C、C	(26)~(30)	C、B、A、D、A
(11)~(15)	C、B、C、D、A	(31)~(35)	C、B、A、D、A
(16)~(20)	A、B、C、C、A		

8.3.3 要点解析

(1) B。单指令流多数据流(SIMD)计算机通常由一个控制器、多个执行部件和多个存

储器模块组成,其典型代表是并行处理机(阵列处理机)。

阵列处理机其工作原理是:将大量重复设置的处理单元互连构成阵列,在单一控制部件的控制下,向各处理单元分配各自的数据,以达到并行执行同一条指令的目的。注意,各处理单元必须以同步方式在同一时间内执行相同的指令。

对于本试题的选项 A,单指令流单数据流(SISD)其实质就是传统的顺序执行的单处理器计算机,其指令部件每次只对一条指令进行译码,并只对一个操作部件分配数据。

选项 C“多指令流单数据流(MISD)”其典型结构具有 n 个处理单元,按 n 条不同指令的要求对同一数据流及其中间结果进行不同的处理。一个处理单元的输出又作为另一个处理单元的输入。这类系统实际上很少见到。

选项 D“多指令流多数据流(MIMD)”是指能实现作业、任务、指令等各级全面并行的多机系统,其典型代表是多处理机。

(2) D。Centronic 总线主要用于计算机与打印机等外设相连接的情况,不能用于多机互联。根据 STD 总线、PCI 总线的规范,它们都支持构成多机系统。而交叉开关就是专门用于多机互联的。

(3) C。该计算机系统的可靠性结构是一个典型的双重串并联结构,而且构成系统的 4 个部件的可靠度均为 $R_1=0.9$,则系统整体的可靠度 $R=1-(1-R_1 \times R_1)(1-R_1 \times R_1)=R_1^2(2-R_1^2)=0.9 \times 0.9 \times (2-0.9 \times 0.9)=0.81 \times 1.19=0.9639$ 。

(4) C; (5) C。依题意知,每一条指令取指时间 $t_{\text{取指}}=5\Delta t$,分析时间 $t_{\text{分析}}=2\Delta t$,执行时间 $t_{\text{执行}}=5\Delta t$ 。如果按顺序方式执行,则每条指令都需三步才能执行完,没有重叠。从头到尾执行完 500 条指令总的执行时间为 $(5+2+5)\Delta t \times 500=6000\Delta t$ 。

流水线技术其实质是一种任务的分解技术。把一件任务分解为若干顺序执行的子任务,不同的子任务由不同的执行机构负责执行,而这些机构可以同时并行工作。在任一时刻,任一任务只占用其中一个执行机构,这样就可以实现多个任务的重叠执行,以提高工作效率。若指令流水线把一条指令分为取指、分析和执行三部分,且三部分的时间分别是 $t_{\text{取指}}=5\Delta t$, $t_{\text{分析}}=2\Delta t$, $t_{\text{执行}}=5\Delta t$,则该流水线的操作周期应取其中时间最长的 $t_{\text{取指}}=5\Delta t$ (或 $t_{\text{执行}}=5\Delta t$)操作部分。

在此指令流水线上执行第 1 条指令是流水线的建立过程,是顺序执行的,故从开始执行第 1 条指令到获得其执行结果所需时间为 3 个基本操作时间之和,即 $(5+2+5)\Delta t=12\Delta t$ 。

对于其余 499 条指令均可在该流水线的操作周期时间($5\Delta t$)内完成。因此,这 500 条指令全部执行完毕所需时间为 $12\Delta t+499 \times 5\Delta t=2507\Delta t \approx 2510\Delta t$,即本题的答案是选项 C。

(6) A。在开发一个系统时,如果用户对系统的目标不很清楚,难以定义需求,这时最好使用原型法的系统开发方法。应用原型法的主要目的就是获取需求。使用原型法,在用户的共同参与下可以改善和加快需求获取过程。其第 1 步是建造一个快速原型,实现客户或未来用户与系统交互,用户或客户对原型进行评价,进一步细化待开发软件的需求。通过逐步调整原型使其满足客户的要求,开发人员可以确定客户的真正需求。第 2 步则在第 1 步的基础上开发客户满意的软件产品。显然,快速原型方法可以克服瀑布模型的缺点,减少由于软件需求不明确带来的开发风险,具有显著的效果。

对于选项 B“瀑布模型”，它是一种将按软件生命周期划分为制定计划、需求分析、软件设计、程序编写、软件测试和运行维护等六个基本活动，并且规定了自上而下、相互衔接的固定次序的系统开发方法。瀑布模型强调文档的作用，并要求每个阶段都要仔细验证。

对于选项 C“V-模型”，它是典型的测试模型。该模型通常会在其开始部分对软件开发过程进行描述，其中通过单元测试检测代码的开发是否符合详细设计的要求；集成测试检测各单元代码是否能完好地结合到一起，是否符合概要设计阶段提出的要求；系统测试检测已集成在一起的产品是否符合系统规格说明书的要求；而验收测试则检测产品是否符合最终用户的需求。

对于选项 D“螺旋模型”，它是指将瀑布模型和快速原型模型结合起来，强调风险分析的一种开发模型。常见的几种系统开发模型优缺点见表 8-14。

表 8-14 常见的几种系统开发模型优缺点

模 型	优 点	缺 点
瀑布模型	文档驱动	系统可能不满足客户的需求
快速原型模型	关注满足客户需求	可能导致系统设计差，效率低，难以维护
增量模型	开发早期反馈及时，易于维护	需要开放式体系结构，可能会导致系统设计差，效率低
螺旋模型	风险驱动	风险分析人员需要有经验且经过充分训练
喷泉模型	对象驱动，使开发过程具有迭代性和无间隙性	

(7) A。测试是保证软件质量的重要手段。根据国家标准 GB 8566—88《计算机软件开发规范》的规定，单元测试是根据详细设计阶段给出的规格说明书在编码阶段完成的测试工作；集成测试的计划是在概要设计阶段制定的；系统测试计划应该在需求分析阶段就开始制定，并在设计阶段细化和完善，而不是等系统编码完成后才制定测试计划；而验收测试则检测产品是否符合最终用户的需求。

软件测试的各个阶段与软件开发阶段的对应关系可以用图 8-8 来表示。

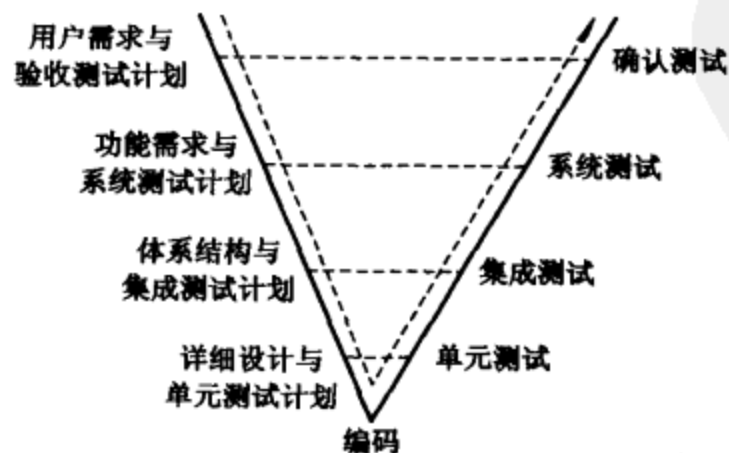


图 8-8 各软件开发阶段与软件测试阶段的对应关系图

(8) C。已经发布实施的标准(包括已确认或修改补充的标准)经过实施一定时期后,对其内容再次审查,以确保其有效性、先进性和适用性。标准的有效期(也称为标龄)是指自标准实施之日起,至标准复审重新确认、修订或废止的时间。由于各国情况不同,因此标准有效期也不同。例如,ISO 标准每 5 年复审一次,平均标龄为 4.92 年。我国国家标准有效期一般为 5 年(即标准实施后的复审周期一般不超过 5 年),有兴趣的读者可参见 1988 年发布的《中华人民共和国标准化法实施条例》中的相关规定。

(9) C。我国著作权法第 2 条规定“中国公民、法人或者其他组织的作品,不论是否发表,依照本法享有著作权”,即著作权不需要登记或标注版权标记就能得到保护。

对于选项 A“专利权”,专利作品的完成人必须依照专利法的有关规定,向国家专利局提出专利申请。专利局依照法定程序进行审查,申请符合专利法规定条件的,由专利局做出授予专利权的决定,颁发专利证书。只有当专利局发布授权公告后,其完成人才享有该项作品的专利权。

对于选项 B“商标权”的获得,我国和大多数国家实行注册制,只有向国家商标局提出注册申请,经审查核准注册后,才能获得商标权。

由于无形的智力创作性成果不像有形财产那样直观可见,因此,确认智力创作性成果的财产权需要依法审查确认才能得到法律保护。

(10) C; (11) C; (12) B。在操作系统的文件系统的树型目录结构中,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。在树型目录结构中,从根目录到任何数据文件之间,只有一条惟一的通路,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惟一的。这样,可以解决文件重名问题。所以,对于第(10)题,虽然问题的研究对象是 f2 数据文件名,但由于其所在的子目录不同,因此子目录 W2 中文件 f2 和子目录 D2 中文件 f2 可能相同也可能不相同,正确答案为选项 C。

从树根开始的路径名为绝对路径名。如果文件系统有很多级,则使用不是很方便,所以引入相对路径名,即从当前目录开始,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。假设当前目录是 D1,进程 A 以“①fd1=open("W1/f1", o_RDONLY);”或“②fd1=open("/D1/W1/f1", o_RDONLY);”两种方式打开文件 f1。

对于方式 1 可用从当前路径开始查找需要访问的文件 f1,但需要访问两次磁盘,第一次查找目录文件 W1,找到文件 f1 的物理位置,第二次便可以读取文件 f1。可见,从当前目录开始,采用相对路径名较之采用绝对路径名,可以减少访问目录文件的次数,工作效率较高。因此,第(11)题的正确答案为 C;第(12)题的正确答案为 B。

(13) C。操作手册主要用于指导用户在系统运行时的操作,系统提供的界面是用户进行各种操作的依据,因此系统处理过程的各个界面应写入操作手册。

系统对各种输入数据的处理方法和系统各部分之间的接口关系是系统开发阶段的内容,系统升级时厂商提供的服务与用户在系统中的操作无关,这些工作不是用户必须关心的内容。

(14) D。代码审查是一种正式的评审活动,而代码走查的讨论过程是非正式的。因此选项 D 说法正确。而对于选项 A 应相应改正为“在代码走查中由编写代码的程序员来组织讨论,而在代码审查中由高级管理人员来领导评审小组的活动”。

对于选项 B 正确的说法是,无论代码审查还是代码走查,都要检查程序与设计文档的一致性。

选项 C 中说要评审程序员的编程能力和工作业绩也是不对的。

(15) A。Gantt 图(因果分析图,又称鱼骨刺图)是一种常用的项目管理工具,它是一种简单的水平条形图,以一个日历为基准描述项目任务,图中横坐标表示时间(如时、天、周、月和年等),纵坐标表示任务,图中的水平线段表示对一个任务的进度安排,线段的起点和终点对应在横坐标上的时间分别表示该任务的开始时间和结束时间,线段的长度表示完成该任务所需的时间。可见,Gantt 图主要用于表现活动的并行性,也可以表现各个活动的起始时间和完成的进度,但不能表现各个活动的顺序和它们之间的因果关系。

(16) A。根据系统开发生命周期可知,系统设计阶段在前,软件编码阶段在后,纠正开发阶段早期的错误要比纠正后期的错误代价大,因此纠正外部和内部设计错误比纠正源代码错误需要更大的成本。

(17) B。一个故障已经被发现,而且也被排除了,因为在修改过程中纠正了旧的错误又会引入新的错误,此时应该进行退化测试(RegesSion Test),以防止出现新的错误。退化测试包括以下步骤:① 插入新代码,程序成为新版本;② 测试可能受新代码影响的功能;③ 测试修改前的基本功能;④ 测试新版本的功能。

选项 A“程序走查”是由 Bell 等人提出来的一种评价程序语言的方法,它以用户为中心,通过走查用户使用程序完成任务的情况进而评价程序语言设计的可用性和流畅性。

选项 C“软件评审”是指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。

选项 D“软件接收测试”用来保证关键任务的应用软件能在所运行的硬件设施上高效地运行,与严格的操作标准一致。

(18) C。假设有一个局域网,管理站每 15 分钟轮询被管理设备一次,一次查询访问需要的时间是 200 ms,管理站最多可以支持 $N = T/\Delta t = (15 \times 60) \text{s} \div 0.2 \text{s} = 4500$ 个网络设备。

(19) C。廉价磁盘冗余阵列(RAID)是利用一台磁盘阵列控制器来管理和控制一组磁盘驱动器,组成一个高度可靠的、快速的大容量磁盘系统。RAID 级别是指磁盘阵列中硬盘的组合方式,不同级别的 RAID 为用户提供的磁盘阵列在性能上和安全性的表现上也有不同,详见表 8-15。

表 8-15 几种 RAID 级别的对比

级别	描 述	优 点	缺 点
RAID0	没有容错设计的条带磁盘阵列,仅提供并行交叉存取功能	有效地提高磁盘 I/O 速度	磁盘系统的可靠性不好
RAID1	具有磁盘镜像和磁盘双工功能	可利用并行读/写特性,将数据块同时写入主盘和镜像盘,故比传统的镜像盘速度快	磁盘利用率只有 50%

续表

级别	描 述	优 点	缺 点
RAID2	增加了汉明码校验与纠错 (Hamming Code ECC) 功能	这是早期为了进行即时数据校验而研制的一种技术, 针对当时对数据安全敏感领域, 如金融服务等	由于花费太大, 成本昂贵, 目前已不用
RAID3	具有并行传输和校验功能	利用一台奇偶校验盘来完成容错功能。比起磁盘镜像, 减少了所需的冗余磁盘数	
RAID4	具有独立的数据硬盘与共享的校验硬盘	与 RAID3 相比, RAID4 是一种相对独立的形式	
RAID5	具有独立的数据磁盘和分布式校验块的磁盘阵列	每个驱动器都有独立的数据通路, 独立地进行读/写, 无专门的校验盘。用于纠错的校验信息是以螺旋方式散布在所有数据盘上。RAID 5 常用于 I/O 较频繁的事务处理上	
RAID6	具有独立的数据硬盘与两个独立的分布式校验方案	设置了一个专用的、可快速访问的异步校验盘。该盘具有独立的数据访问通路	其性能改进有限, 价格却很昂贵
RAID7	具有最优化的异步高速率 I/O 和高数据传输率的磁盘阵列	是对 RAID6 级的改进。在这种阵列中的所有磁盘, 都具有较高的传输速度, 有着优异的性能, 是目前最高档次的磁盘阵列	
RAID10	建立在 RAID0 和 RAID1 基础上的高可靠性与高性能的组合	利用了 RAID0 极高的读写效率和 RAID1 较高的数据保护和恢复能力, 使 RAID10 成为了一种性价比较高的等级, 目前几乎所有的 RAID 控制卡都支持该等级	

(20) A. 流媒体技术是指把连续的影像和声音信息经过压缩处理之后放到专用的流服务器上, 让浏览者一边下载一边观看、收听, 而不需要等到整个多媒体文件下载完成就可以即时收看和收听的技术。

流媒体系统在服务器端完成流媒体文件的压缩生成, 经过服务器发布, 然后在客户端完成流媒体文件的解压播放。流媒体影音数据的传输一般采用建立在用户数据报协议 (UDP) 上的实时传输协议。

(21) D。代理服务器是指在计算机客户端和访问的计算机网络(如 Internet 网)之间安装有相应代理服务器软件的一台计算机,客户端对网络的所有访问请求都通过代理服务器实现。而被访问的网络计算机对请求的回答,也通过代理服务器转达到客户端。代理服务器的作用主要有以下 4 个:

① 代理服务器提供远程信息本地缓存功能,减少信息的重复传输。

② 所有使用代理服务器的用户都必须通过代理服务器访问远程站点,因此在代理服务器上就可以设置相应的限制,以过滤或屏蔽掉某些信息。因此代理服务器可以起到防火墙的作用。

③ 通过代理服务器可访问一些不能直接访问的网站。互联网上有许多开放的代理服务器,客户在访问权限受到限制时,而这些代理服务器的访问权限是不受限制的,刚好代理服务器在客户的访问范围之内,那么客户通过代理服务器访问目标网站就成为可能。

④ 安全性得到提高。无论是上聊天室还是浏览网站,目的网站只能知道你来自于代理服务器,而你的真实 IP 地址就无法测知,这就使得使用者的安全性得以提高。

对于选项 D“信息加密”,通常要使用专用的加密软件及加密算法来完成。

(22) A。地址解析协议(ARP)是数据链路层协议,但同时对上层(网络层)提供服务,完成将 IP 地址转换成以太网的 MAC 地址的功能。

ARP 工作时,送出一个含有所希望的 E 地址的以太网广播数据包。当发出 AM 请求时,发送方填好发送方首部和发送方 IP 地址,还要填写目标 E 地址。当目标机器收到这个 ARP 广播帧时,就会在响应报文中填上自己的 48 位主机地址。由此可以看出,ARP 广播帧最初是以 IP 地址的形式来寻址发送的,所以需要工作在网络层的网络设备路由器对其进行隔离。可见路由器能完成“隔离冲突域,隔离广播域”的功能。

网桥、以太网交换机属于数据链路层的网络互联设备,完成“隔离冲突域,共享广播域”的功能。

集线器属于物理层的网络互联设备,完成“共享冲突域,共享广播域”的功能。

(23) B。RAS(Remote Access Service, 远程访问服务组件)是微软公司在 Windows 系统中提供的,它允许用户通过拨号方式登录网络系统,从而实现移动办公,即无论出差在外还是下班回家,职工希望随时登录到公司的网络中查看、下载资料,或者为客户提供登录网络查询业务数据服务。

IIS(Internet Information Server, 互联网信息服务)是一个可以支持 HTTP、FTP 以及 SMTP 等协议,运行在一台装有 Windows 操作系统的计算机上,实现 Internet 或 Intranet 的 Web、FTP 等站点服务功能的软件。

DNS 域名解析服务系统是一种 TCP/IP 的标准服务。DNS 服务允许网络上的客户机注册和解析 DNS 域名。这些名称用于为搜索和访问网络上的计算机提供定位。DNS 域名解析服务,可实现域名与 IP 地址之间的转换功能,把域名转换成与其相对应的 IP 地址,也可以把 E 地址反向解析转换成域名,以解决 IP 地址难以记忆的问题。

而超文本传输协议(HTTP)是一个应用于 WWW 信息服务系统的应用层协议,它不属于 Windows 系统的网络服务组件。

(24) B。OSI 参考模型在每一层都定义了实现的功能、数据传输格式和服务访问点,如表 8-16 所示。

表 8-16 OSI 参考模型各层功能、数据单元和 SAP

层次	主要功能	数据单元	服务访问点(SAP)
应用层	语义转换,实现多个进程相互通信的同时,完成一系列业务处理所需的服务功能	报文	用户界面
表示层	语法转换,给用户提供一种执行会话服务的方式,管理当前请求数据结构组		
会话层	数据交换,对话管理、会话同步		
传输层	实现端-端系统无差错的和有序的报文收发		端口号
网络层	提供路由选择,并具有拥塞控制、信息包顺序控制及网络记账等功能	分组	IP 地址
数据链路层	实现系统实体间透明的帧传输,并为网络层提供可靠无错误的数据信息	帧	MAC 地址
物理层	通过机械、电气、功能、规程等特性实现实体之间正确的、透明的位流传输	比特	网卡接口

(25) B。OGSA 是开放网络服务框架,是 Web Service 和 Grid 技术结合的产物,已成为网络基础框架的标准。它利用 Web Service 的标准接口定义机制、多协议绑定、本地与远端的透明性,利用 Grid 的服务语义、可靠性和安全模型、生命周期管理、发现和其他服务,以及多主机或运行环境来建构自己的框架。为了使服务的思想更加明确和具体,OGSA 定义了网络服务(Grid Service)的概念,用于解决服务的发现、动态服务的创建、服务生命周期的管理与临时服务有关的问题。Grid Service 提供了一组遵守特定的约定并定义了明确的接口,是实体之间产生、管理和交换信息的机制。

选项 A“Object”是指面向对象技术中的对象。

选项 C“Web Service”是由 W3C 协会设计和指定,用来促进跨平台的程序对程序通信的一系列标准。目前,W3C 已指定一个模板 WSDL(Web 服务描述语言)和一个过程调用协议 SOAP(简单对象访问协议)的编程接口作为正式的 Web Service 标准。

选项 D“XML”是可扩展置标语言的英文缩写,用于标记电子文件的结构化语言,允许用户自己定义一套适合于应用的文档元素类型,具有较高的灵活性。

(26)~(30) C、B、A、D、A。

参考译文: MIDI 使得人们能够使用多媒体计算机和电子乐器。实际上 MIDI 有 3 种元素:通信协议、硬件接口和被称为标准 MIDI 文件的发布格式。在 WWW 环境中,最重要的元素是音频格式。在原理上, MIDI 文件包含了 MIDI 协议报文序列。然而,当 MIDI 协议报文被存储在 MIDI 文件中时,事件都被加上了时间戳,以便于按照适当的顺序回放。由 MIDI 文件提供的音乐是 MIDI 当前最通常的应用。

(31)~(35) C、B、A、D、A。

参考译文: 证书是一种数字文档,用于表明把一个公钥绑定到一个人或其他实体。用它验证一个给定的公钥确实属于某一个人。证书可以防止某些人使用假冒的密钥去冒

充别人。最简单的证书包含一个公钥和一个名字。通常使用的证书也包含超时日期、发行证书的 CA 的名字、一个序列号以及其他信息。最重要的是,它包含了证书发行者的数字签名。最广泛接受的证书格式是 X.509,这样的证书可以被任何服从 X.509 标准的应用读或写。

8.4 2006 年上半年真题链接

8.4.1 上午试题

• 两个同符号的数相加或异符号的数相减,所得结果的符号位 SF 和进位标志 CF 进行(1)运算为 1 时,表示运算的结果产生溢出。

(1) A. 与 B. 或 C. 与非 D. 异或

• 浮点数的阶码用移码表示,尾数用补码表示。两规格化浮点数相乘,最后对结果规格化时,右规的右移位数最多为(2)位。

(2) A. 1 B. 2 C. 尾数位数 D. 尾数位数-1

• 高速缓存 Cache 与主存间采用全相联地址映像方式,高速缓存的容量为 4 MB,分为 4 块,每块 1 MB,主存容量为 256 MB。若主存读写时间为 30 ns,高速缓存的读写时间为 3 ns,平均读写时间为 3.27 ns,则该高速缓存的命中率为(3)%。若地址变换表如表 8-17 所示,则主存地址为 8888888H 时,高速缓存地址为(4)H。

表 8-17 地址变换表

0	38H
1	88H
2	59H
3	67H

(3) A. 90 B. 95 C. 97 D. 99

(4) A. 488888 B. 388888 C. 288888 D. 188888

• 若某计算机系统是由 500 个元器件构成的串联系统,且每个元器件的失效率均为 $10^{-7}/h$,在不考虑其他因素对可靠性的影响时,该计算机系统的平均故障间隔时间为(5)小时。

(5) A. 2×10^4 B. 5×10^4 C. 2×10^5 D. 5×10^5

• 某指令流水线由 5 段组成,各段所需要的时间如图 8-9 所示。连续输入 10 条指令时的吞吐率为(6)。

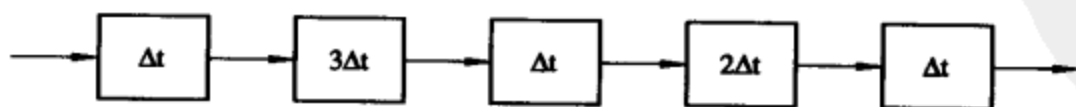


图 8-9 某指令流水线各段所需要时间图

- (6) A. $10/70\Delta t$ B. $10/49\Delta t$ C. $10/35\Delta t$ D. $10/30\Delta t$

• 相对于 DES 算法而言, RSA 算法的(7), 因此, RSA(8)。

- (7) A. 加密密钥和解密密钥是不相同的
B. 加密密钥和解密密钥是相同的
C. 加密速度比 DES 要高
D. 解密速度比 DES 要高

- (8) A. 更适用于对文件加密
B. 保密性不如 DES
C. 可用于对不同长度的消息生成消息摘要
D. 可以用于数字签名

• 驻留在多个网络设备上的程序在短时间内同时产生大量的请求消息冲击某 Web 服务器, 导致该服务器不堪重负, 无法正常响应其他合法用户的请求, 这属于(9)。

- (9) A. 网上冲浪 B. 中间人攻击
C. DDoS 攻击 D. MAC 攻击

• 上海市标准化行政主管部门制定并发布的工业产品的安全、卫生要求的标准, 在其行政区域内是(10)。

- (10) A. 强制性标准 B. 推荐性标准
C. 自愿性标准 D. 指导性标准

• 小王购买了一个“海之久”牌移动硬盘, 而且该移动硬盘还包含有一项实用新型专利, 那么, 王某享有(11)。

- (11) A. “海之久”商标专用权 B. 该盘的所有权
C. 该盘的实用新型专利权 D. 前三项权利之全部

• MPC(Multimedia PC)与 PC 的主要区别是增加了(12)。

- (12) A. 存储信息的实体 B. 视频和音频信息的处理能力
C. 光驱和声卡 D. 大容量的磁介质和光介质

• 人眼看到的任一彩色光都是亮度、色调和饱和度三个特性的综合效果, 其中(13)反应颜色的种类。

- (13) A. 色调 B. 饱和度 C. 灰度 D. 亮度

• CD 上声音的采样频率为 44.1 kHz, 样本精度为 16 b/s, 双声道立体声, 那么其未经压缩的数据传输率为(14)。

- (14) A. 88.2 kb/s B. 705.6 kb/s
C. 1411.2 kb/s D. 1536.0 kb/s

• 在软件项目管理中可以使用各种图形工具来辅助决策, 下面对 Gantt 图的描述中, 不正确的是(15)。

- (15) A. Gantt 图表现了各个活动的持续时间
B. Gantt 图表现了各个活动的起始时间
C. Gantt 图反映了各个活动之间的依赖关系
D. Gantt 图表现了完成各个活动的进度

• 模块的耦合度描述了(16)。

- (16) A. 模块内各种元素结合的程度 B. 模块内多个功能之间的接口
C. 模块之间公共数据的数量 D. 模块之间相互关联的程度

• 内聚是一种指标,表示一个模块(17)。

- (17) A. 代码优化的程度
B. 代码功能的集中程度
C. 完成任务的及时程度
D. 为了与其他模块连接所要完成的工作量

• 在软件项目开发过程中,评估软件项目风险时,(18)与风险无关。

- (18) A. 高级管理人员是否正式承诺支持该项目
B. 开发人员和用户是否充分理解系统的需求
C. 最终用户是否同意部署已开发的系统
D. 开发需要的资金是否能按时到位

• 开发专家系统时,通过描述事实和规则由模式匹配得出结论,这种情况下适用的开发语言是(19)。

- (19) A. 面向对象语言 B. 函数式语言
C. 过程式语言 D. 逻辑式语言

• 高级程序设计语言中用于描述程序中的运算步骤、控制结构及数据传输的是(20)。

- (20) A. 语句 B. 语义 C. 语用 D. 语法

• 为了解决进程间的同步和互斥问题,通常采用一种称为(21)机制的方法。若系统中有5个进程共享若干个资源R,每个进程都需要4个资源R,那么使系统不发生死锁的资源R的最少数目是(22)。

- (21) A. 调度 B. 信号量 C. 分派 D. 通讯
(22) A. 20 B. 18 C. 16 D. 15

• 在UNIX操作系统中,把输入/输出设备看做是(23)。

- (23) A. 普通文件 B. 目录文件 C. 索引文件 D. 特殊文件

• 某软盘有40个磁道,磁头从一个磁道移至另一个磁道需要5 ms。文件在磁盘上非连续存放,逻辑上相邻数据块的平均距离为10个磁道,每块的旋转延迟时间及传输时间分别为100 ms和25 ms,则读取一个100块的文件需要(24)时间。

- (24) A. 17 500 ms B. 15 000 ms C. 5000 ms D. 25 000 ms

• 文件系统中,设立打开文件(Open)系统功能调用的基本操作是(25)。

- (25) A. 把文件信息从辅存读到内存
B. 把文件的控制管理信息从辅存读到内存
C. 把磁盘的超级块从辅存读到内存
D. 把文件的FAT表信息从辅存读到内存

• 与多模光纤相比较,单模光纤具有(26)等特点。

- (26) A. 较高的传输率、较长的传输距离、较高的成本
B. 较低的传输率、较短的传输距离、较高的成本
C. 较高的传输率、较短的传输距离、较低的成本
D. 较低的传输率、较长的传输距离、较低的成本

表 8-18 参考答案及评分标准表

问题号	参考答案	问题号	参考答案
(1)~(5)	D、A、D、D、A	(21)~(25)	B、C、D、A、B
(6)~(10)	C、A、D、C、A	(26)~(30)	A、C、A、C、B
(11)~(15)	B、B、A、C、C	(31)~(35)	B、C、A、D、B
(16)~(20)	D、B、C、D、A	(36)~(40)	A、C、B、A、D

8.4.3 要点解析

(1) D。判别两个同符号的数相加或异符号的数相减是否溢出的逻辑表达式是： $VF = SF \oplus CF$ ，即将所得结果的符号位 SF 和进位标志 CF 进行异或运算，结果为 1 时表示溢出，反之表示没有溢出。

(2) A。通常浮点数的阶码用移码表示，尾数用补码表示。由于规格化浮点数的尾数的取值范围为： $[1/2]_{\text{补}} \leq [M]_{\text{补}} < [1]_{\text{补}}$ ，或 $[-1]_{\text{补}} \leq [M]_{\text{补}} < [-1/2]_{\text{补}}$ ，因此将两个尾数相乘，积的取值范围为： $[1/4]_{\text{补}} \leq [M_{\text{积}}]_{\text{补}} \leq [1]_{\text{补}}$ ，或 $[-1]_{\text{补}} < [M_{\text{积}}]_{\text{补}} < [-1/2]_{\text{补}}$ 。两规格化浮点数相乘，最后对结果规格化时，右规的右移位数最多为 1 位。

(3) D；(4) D。设命中率为 x ，依题意“若主存读写时间为 30 ns，高速缓存的读写时间为 3 ns，平均读写时间为 3.27 ns”得： $3x + 30(1-x) = 3.27$ 。解得： $x = 0.99$ ，即该 4 MB 高速缓存的命中率为 99%。

高速缓存 Cache 与主存间采用全相联地址映像方式，其中“全相联地址映像方式”是指主存的每一页可以映像到 Cache 的任意一页。由于 1 MB 的十六进制数表达范围是 000000H~0FFFFFFH，因此可将地址变换表(表 8-17)理解成表 8-19 的形式。

表 8-19 地址变换表

Cache 地址范围	内存地址范围
000000H~0FFFFFFH	3800000H~38FFFFFFH
100000H~1FFFFFFH	8800000H~88FFFFFFH
200000H~2FFFFFFH	5900000H~59FFFFFFH
300000H~3FFFFFFH	6700000H~67FFFFFFH

查表 6-19 可知，主存地址为 8888888H 时，高速缓存地址为 188888H。

另一种解答思路是：当主存地址为 8888888H 时，即二进制地址为 1000100010001000100010001000B，其中块内地址为 1000100010001000B，而相联存储器中存储的是区号 100010B 和区内块号 00B，也就是相联存储器中存储的是 10001000B=88H。由相联存储器的 88H 查出 Cache 块号为 01B。将 Cache 块号与块内地址连接到一起，构成 Cache 的地址为 0110001000100010001000B 即 188888H。

(5) A。根据题意，由 500 个元器件构成的串联计算机系统的总失效率为各元器件的失效率的和，即为 $500 \times 10^{-7}/h = 5 \times 10^{-5}/h$ 。

在不考虑其他因素对可靠性的影响时,由于计算机系统的平均故障间隔时间可定义为该系统失效率的倒数,因此该计算机系统的平均故障间隔时间(MTBF)为 $2 \times 10^4 \text{h}$ 。

(6) C。从图 8-9 可看出,该流水线各段所经历的时间不一样,因此该指令流水线吞吐率的计算公式为

$$TP = \frac{n}{\sum_{i=1}^m \Delta t_i + (n-1)\Delta t_j}$$

式中: m 为流水线的段数; n 为输入的指令数; Δt_i 为第 i 段所需时间; Δt_j 为该段流水线中瓶颈段的时间。将题中已知条件代入上式,求得吞吐率 TP 为

$$\begin{aligned} TP &= \frac{n}{\sum_{i=1}^m \Delta t_i + (n-1)\Delta t_j} \\ &= \frac{10}{(1+3+1+2+1)\Delta t + (10-1) \times 3\Delta t} \\ &= \frac{10}{35\Delta t} \end{aligned}$$

(7) A; (8) D。DES 是一种对称密钥密码算法,它的加密密钥和解密密钥是相同的。而 RSA 是一种非对称密钥密码算法,它使用不同的密钥分别用于加密和解密数据,还可以用于数字签名。通常对称密钥密码算法的效率要比非对称密钥密码算法高很多,适用于对文件等大量的数据进行加密。

(9) C。DDoS(分布式拒绝服务)的英文全称为 Distributed Denial of Service,它是一种基于 DoS 的特殊形式的拒绝服务攻击,是一种分布、协作的大规模攻击方式,多个网络设备上的程序在短时间内同时向某台服务器产生大量的请求,导致该服务器不堪重负,无法正常响应其他合法用户的请求。

(10) A。《标准化法》第 2 章第 7 条规定“国家标准、行业标准分为强制性标准和推荐性标准。保障人体健康,人身、财产安全的标准和法律、行政法规规定强制执行的标准是强制性标准,其他标准是推荐性标准。”

“省、自治区、直辖市标准化行政主管部门制定的工业产品的安全、卫生要求的地方标准,在本行政区域内是强制性标准。”

由以上分析可知,按照我国《标准化法》的规定,上海市标准化行政主管部门制定并发布的工业产品的安全、卫生要求的标准,在其行政区域内是强制性标准。

(11) B。选项 A“商标专用权”是指企业、事业单位和个体工商户,对其生产、制造、加工、拣选或者经销的商品,向商标局申请商品商标注册,经商标局核准注册的商标为注册商标,所取得的专用权受法律保护。并且,促使生产者、制造者、加工者或经销者保证商品质量和维护商标信誉,对其使用注册商标的商品质量负责,便于各级工商行政管理部门通过商标管理,监督商品质量,制止欺骗消费者的行为。

选项 C“实用新型专利权”是指受我国《专利法》保护的发明创造权利。实用新型专利权被授予后,除法律另有规定的以外,任何单位或者个人未经专利权人许可,不得以生产经营为目的制造、使用、销售其专利产品,或者使用其专利方法以及使用、销售依照该专利方法直接获得的产品。由此可见,小王购买了“海之久”牌移动硬盘,只享有该移动硬盘的

所有权,而不享有题目中所提及移动硬盘的其他权利。

(12) B。多媒体是融合两种或者两种以上媒体的一种人-机交互式信息交流和传播媒体,使用的媒体包括文字、图形、图像、声音、动画和电视图像等。

要把一台普通的计算机变成多媒体计算机(MPC)需研究的关键技术是:视频、音频信号获取技术、多媒体数据压缩编码和解码技术、视频、音频数据的实时处理技术和视频、音频数据的输出技术。可见多媒体计算机主要是在普通计算机的基础上增加了视频和音频信息的处理能力。

(13) A。为了能确切地表示某一彩色光的度量,可以用亮度、色调和色饱和度3个物理量来描述,并称之为色彩三要素。

亮度:亮度是描述光作用于人眼时引起的明暗程度感觉,是指色彩明暗、深浅程度。

色调:色调是指颜色的类别,如红色、绿色、蓝色等不同颜色就是指色调。

色饱和度:色饱和度是指某一颜色的深浅程度(或浓度)。

(14) C。波形声音信息是一个用来表示声音振幅的数据序列,它是通过对模拟声音按一定间隔采样获得的幅度值,再经过量化和编码后得到的便于计算机存储和处理的数据格式。

$$\begin{aligned}\text{未经压缩的数字音频数据传输速率(b/s)} &= \text{采样频率(Hz)} \times \text{量化位数(b)} \times \text{声道数} \\ &= 44.1 \text{ kHz} \times 16 \text{ b/s} \times 2 = 1411.2 \text{ kb/s}\end{aligned}$$

(15) C。PERT(项目评估与评审技术)图和 Gantt 图(因果分析图,又称鱼骨刺图)是两种常用的项目管理工具。其中,PERT 图是一种图形化的网络模型,描述一个项目中任务与任务之间的依赖关系。

Gantt 图是一种简单的水平条形图,它以一个日历为基准描述项目任务。Gantt 图中的横坐标表示时间(如时、天、周、月和年等),纵坐标表示任务,图中的水平线段表示对一个任务的进度安排,线段的起点和终点对应在横坐标上的时间分别表示该任务的开始时间和结束时间,线段的长度表示完成该任务所需的时间。由此可见,Gantt 图只能表示任务之间的并行与串行的关系,难以反映多个任务之间存在的复杂关系,不能直观表示任务之间相互依赖制约关系,以及哪些任务是关键任务等信息。

(16) D。模块的耦合度表现了模块之间相互关联的程度,从低到高分为无直接耦合、数据耦合、标记耦合、控制耦合、公共耦合和内容耦合六级。

(17) B。模块的内聚性表现了模块内部各个子成分(代码)功能的集中程度,从低到高分偶然内聚、逻辑内聚、时间内聚、通信内聚、顺序内聚和功能内聚六级。

(18) C。在软件项目开发过程中,软件项目风险与高级管理人员的支持程度有关,与对系统需求理解的程度有关,与开发资金的及时投入有关,但是与最终用户无关,系统的最后部署与运行不属于开发过程。

(19) D。用逻辑式程序设计语言编写程序不需要描述具体的解题过程,只需要给出一些必要的事实和规则。这些规则是解决问题的方法的规范说明,根据这些事实和规则,计算机利用谓词逻辑,通过演绎推理得到求解问题的执行序列。这种语言主要用在人工智能领域,也应用在自然语言处理、数据库查询、算法描述等方面,尤其适合于作为专家系统的开发工具。

对于选项 B“函数式语言”,函数式程序设计的数据结构本质上是表,而函数又可以作

为值出现在表中,因此函数式程序的控制结构取决于函数,以及函数的定义和调用。函数式语言主要用于符号数据处理,如微分和积分演算、数理逻辑、游戏推演以及人工智能等其他领域。

(20) A。在高级程序设计语言中,语句用于描述程序中的运算步骤、控制结构及数据传输。

程序设计语言的语法是语言的外观。给出语言的语法意味着给出语句、声明和其他语言结构的书写规则。语义则表示不同的语法结构的含义。在程序语言的手册中,语言的描述都是围绕着语法结构展开的。通常,先给出各种语句结构的语法,然后给出对应该结构的语义以描述内在含义。

语用表示关于程序与使用者之间的关系。

(21) B; (22) C。由于在系统中,多个进程竞争同一资源可能会发生死锁,若无外力作用,这些进程都将永远不能再向前推进。为此,在操作系统的进程管理中最常用的方法是采用信号量(Semaphore)机制。

信号量是表示资源的实体,是一个与队列有关的整型变量,其值仅能由 P、V 操作改变。“P 操作”用来检测信号量是否为正值,若不是,则阻塞调用进程;“V 操作”用来唤醒一个阻塞进程恢复执行。

根据用途不同,信号量分为公用信号量和私用信号量。公用信号量用于实现进程间的互斥,初值通常设为 1,它所联系的一组并行进程均可对它实施 P、V 操作;私用信号量用于实现进程间的同步,初始值通常设为 0 或 n。

若系统中有 5 个进程共享若干个资源 R,每个进程都需要 4 个资源 R,假设系统为每个进程各分配了 3 个资源,即 5 个进程共分配了 15 个单位的资源 R,此时只要再有 1 个资源 R,就能保证有一个进程运行完毕。当该进程释放其占有的所有资源时,其他进程又可以继续运行,直到所有进程运行完毕。因此,使系统不发生死锁的资源 R 的最少数目是 16。

(23) D。在 UNIX 操作系统中,把输入/输出设备看做是特殊文件。在 UNIX 系统中包括两类设备:块设备和字符设备。设备特殊文件有一个索引节点,在文件系统目录中占据一个节点,但其索引节点上的文件类型与其他文件不同,是“块”或者“字符”特殊文件。

(24) A。访问一个数据块的时间=寻道时间+旋转延迟时间+传输时间。根据题意,每块的旋转延迟时间及传输时间共需 $100+25=125$ ms,磁头从一个磁道移至另一个磁道需要 5 ms,但逻辑上相邻数据块的平均距离为 10 个磁道,即读完一个数据块到下一个数据块的寻道时间需要 50 ms。通过上述分析,本题访问一个数据块的时间应为 $125+50=175$ ms。当需要读取一个 100 块的文件时,则共需要 17 500 ms 的时间。

(25) B。在使用已经存在的文件之前,要通过“打开(Open)”文件操作建立起文件和用户之间的联系,目的是把文件的控制管理信息从辅存读到内存。打开文件应完成如下功能:

① 在内存的管理表中申请一个空表目,用来存放该文件的文件目录信息。

② 根据文件名在磁盘上查找目录文件,将找到的文件目录信息复制到内存的管理表中。如果打开的是共享文件,则应进行相关处理,如共享用户数加 1。

③ 文件定位,卷标处理等。

文件一旦打开,可被反复使用直至文件关闭。这样做的优点是减少查找目录的时间,

加快文件存取速度,提高系统的运行效率。

(26) A。多模光纤的特点是成本低,芯线宽,聚光好,耗散大且效率低,用于低速度、短距离的通信。

单模光纤的特点是成本高,芯线窄,需要激光源,耗散小且效率高,用于高速度、长距离的通信。

(27) C; (28) A。码分多址(CDMA, Code Division Multiple Access)技术比较适合现代移动通信网的大容量、高质量、综合业务、软切换等要求。CDMA 技术的原理是基于扩频技术,即将需传送的具有一定信号带宽的信息数据用一个带宽远大于信号带宽的高速伪随机码进行调制,使原数据信号的带宽被扩展,再经载波调制并发送出去。接收端使用完全相同的伪随机码,与接收的带宽信号作相关处理,把宽带信号换成原信息数据的窄带信号,即解扩,以实现信息通信。

2006 年 2 月,信息产业部在北京宣布,TD-SCDMA 标准为我国通信行业标准。该标准是由中国大陆独自制定的 3G 标准,1999 年 6 月 29 日,中国原邮电部电信科学技术研究院(大唐电信)向 ITU 提出。该标准将智能无线、同步 CDMA 和软件无线电等当今国际领先技术融于其中,在频谱利用率、对业务支持的灵活性、频率灵活性及成本等方面具有独特优势。

(29) C。“`<title style="italic">science</title>`”是一个 XML 元素的定义。其中, title 是元素标记的名称; style 是元素标记的属性名称; italic 是元素标记的属性值; science 是元素内容。

(30) B。当网络无法访问外部站点时,管理人员在 Windows 操作系统下可以使用 tracert 跟踪网络连接。tracert(跟踪路由)是路由跟踪实用程序,用于确定 IP 数据包访问目标所采取的路径。通过该命令可以查看在哪段路由出现连通问题,从而判断故障发生在校园网内还是校园网外。

如果采用 ping 操作,则只能判断用户与外部站点的连通性,但是无法判断故障处于校园网内还是校园网外。

netstat 用于显示与 IP、TCP、UDP 和 ICMP 协议相关的统计数据,一般用于检验本机各端口的网络连接情况。题目中“netstat 210.102.58.74”的命令格式是错误的。

使用 arp 命令可以查看和修改本地计算机上的 arp 表项。该命令对于查看 arp 缓存和解决地址解析问题非常有用。

(31)~(35) B、C、A、D、B。

参考译文: Cookies 原来是由 Netscape 通信公司引入的,这是 HTTP 服务器方应用程序的一种通用机制,就像 CGI 脚本一样,它可以由 HTTP 连接的客户端用于存储和检索信息。Cookies 的基本功能是弥补 HTTP 无状态的缺陷。它能够通过简单而持续地维护客户端的状态来扩展基于 WWW 的应用能力。

(36)~(40) A、C、B、A、D。

参考译文: WebSQL 是一种类似于 SQL 的查询语言,用于从 Web 中提取信息。它能够在 Web 超文本中巡航,这使得它成为自动操作一个页面中有关链接的有用工具,或是作为搜索从一个给定的 URL 可以到达的、所有匹配某种模式的页面的有用工具。WebSQL 也提供透明地访问索引服务器的手段,这种服务器可以通过公共网关接口进行查询。

8.5 2006 年下半年真题链接

本次考试上午试卷公共试题(1)~(22),请参见本书第7章“2006 年下半年嵌入式系统设计师试卷”上午试卷试题(1)~(22)。



参 考 文 献

本书在编写过程中,参考了许多相关的书籍和因特网上的资料,在此对这些参考文献的作者表示感谢。

1. 主要参考的书籍

- [1] 全国计算机技术与软件专业技术资格(水平)考试办公室. 2006 年嵌入式系统设计师考试大纲. 北京:清华大学出版社, 2006
- [2] 魏洪兴. 嵌入式系统设计师教程. 北京:清华大学出版社, 2006
- [3] 全国计算机技术与软件专业技术资格(水平)考试办公室. 2006 年下半年嵌入式系统设计师考试试题
- [4] 全国计算机技术与软件专业技术资格(水平)考试办公室. 2004 年下半年试题分析与解答. 北京:清华大学出版社, 2005
- [5] 全国计算机技术与软件专业技术资格(水平)考试办公室. 2005 年上半年试题分析与解答. 北京:清华大学出版社, 2005
- [6] 全国计算机技术与软件专业技术资格(水平)考试办公室. 2005 年下半年试题分析与解答. 北京:清华大学出版社, 2006
- [7] 全国计算机技术与软件专业技术资格(水平)考试办公室. 2006 年上半年试题分析与解答. 北京:清华大学出版社, 2006
- [8] 全国计算机技术与软件专业技术资格(水平)考试办公室. 1996~2006 年系统分析师考试试题
- [9] 全国计算机技术与软件专业技术资格(水平)考试办公室. 2004~2006 年软件设计师考试试题
- [10] 全国计算机技术与软件专业技术资格(水平)考试办公室. 2001~2006 年网络工程师考试试题
- [11] 贾智平. 嵌入式系统原理与接口技术. 北京:清华大学出版社, 2006
- [12] 魏洪兴. 嵌入式系统设计与实例开发实验教材 I——基于 ARM 微处理器与 $\mu\text{C}/\text{OS-II}$ 实时操作系统. 北京:清华大学出版社, 2005
- [13] 魏洪兴. 嵌入式系统设计与实例开发实验教材 II——基于 ARM9 微处理器与 Linux 操作系统. 北京:清华大学出版社, 2005
- [14] 王福瑞. 单片微机测控系统设计大全. 北京:北京航空航天大学出版社, 1998
- [15] 郭春柱. 网络工程师考试案例导学. 北京:人民邮电出版社, 2006

2. 主要参考的网站

- [1] <http://www.ceiaec.org>(信息产业部电子教育与考试中心)
- [2] <http://jpk.hrbust.edu.cn/wjjs>(微型计算机系统与接口技术)
- [3] <http://www.infonet.ustc.edu.cn>(信息网络实验室)
- [4] <http://www.chinaitlab.com>(中国 IT 实验室)
- [5] <http://www.chinalinuxpub.com>(Linux 技术中坚站)