现代电子计算机有处理能力，主要的贡献是香农

电子半导体计算载体发热意味着计算系统整体效率低

普适计算愿景符合主流专用计算系统，普适计算是把计算溶解到对象、环境之中

通过嵌入式系统定义解析，就合理代价全面描述构成嵌入式系统所需要的核心器件资源、体积、成本、能耗、工期等；

下面关于组织型嵌入式系统与非组织型嵌入式系统的描述组织型嵌入式系统内固有存在应用周期采集、更新安全可信机制，非组织型的没有

下面关于都江堰水利工程的说法哪些是正确的

都江堰工程改变了中国历史的走向

都江堰工程是一个活的，有生命的工程，农业灌溉从开始的 100 万亩，到现在的 1000 亩

都江堰工程支持了道教文化的发源

数学模型、控制数学模型定义与嵌入式系统定义的相似之处描述数学模型与控制数学模型是目标驱动定义的,数学手段仅是工具,这逻辑与嵌入式系统类似。

下面有关 CISC 和 RISC 指令集的描述RISC 并发能力强于 CISC

关于嵌入式系统硬件体系的功耗设计通常系统能耗越低越好，能耗与系统性能需要妥协均衡；系统性能与功耗是互斥的

为什么嵌入式硬件系统时钟需要 RTC 与系统时钟两类，而系统时钟又通过相对独立 的 PLL 形成多套系统时钟？

RCT 是为了系统睡眠唤醒而设立；

PLL 是为了降低外部时钟振荡器频率；

多个 PLL 组成的多系统时钟，可以分别协同系统、总线、外设及内容不同等级速度的需求

有关 Zigbee 网路描述ZigBee 精简功能设备支持星型网络

关于嵌入式系统的传感与执行器部件描述MEMS 陀螺仪 属于传感器

开源硬件的描述RISC-V 是开源的，Arduinio 是开源，树莓派不是全开源的

ARM 芯片单纯授权制造收 license 和 royalty 的最大难处是核心技术过硬，融合包容性强

ARM 架构 Vs 处理器交织滚动发展的描述ARM 架构是 ARM 处理器的抽象，ARM 处理器是 ARM 结构的具象；

ARM 处理器异常在计算架构上价值是是对出现计算处理运行问题的细分应对，ARM 处理器中复位是最高级的异常，软中断异常是最低级别的异常，ARM 处理器异常是对传统中断的扩充

ARM cortext-A 系列处理器的存储映射中，为什么需要匹配那么多种类存储器MMU 具有 MPU 存取保护功能

课设实验箱中的 TF 卡引导，属于 iROM 引导

有关设备驱动程序步骤的描述接受上层入软件发送过来的抽象要求，发送具体要求给设备控制器，启动设备去执行，执行设备中断程序，I/O 处理完成后的收尾工作

关于嵌入式系统引导 UBOOT stage1 与 stage2 描述哪些是正确的？uboot 是开源的，进入 uboot 状态后可以使用后一些命令进行设置、传输、填充数据等

uboot 通过编译配置文件 board.cfg 完成相应硬件核心板的定制。

安卓电磁翻盖通过驱动程序接入系统应用，基本流程描述

翻盖打开或闭合，通过霍尔传感控制器，产生引脚电平变化；

由驱动层产生 uevent 事件，然后接入 framework 层

应用层接入接受已分发事件并处理

iRomboot 为最小系统 I；

**关于嵌入式 OS 层次内核与微内核**

**层次与微内核是两种基本的嵌入式操作系统构成形态**

**微内核操作系统更加适合嵌入式系统的裁剪**

**关于开源、自由软件，及其软件许可证，GPL 是自由软件的软件许可证；**

**有关 linuxkernel 开源，linuxkernel 的编写语言主要是 C 语言**

**嵌入式系统的实时性能描述**

**系统的实时性是指响应时间既要快，又要在规定的时间内获得正确的输入，及施加正确的输出；**

**三维测量方法是通过处理器的综合性能描述系统的实时性能。**

**μCOS&QNX&VXworks 描述 Vxworks 系统具备实时可预测性与性能的公平性；**

**关于物联 OS 的描述**

**物联 OS 通常具备更加微小，并且具有较强的节点互联能力；**

**有关安卓架构的描述**

**安卓硬件抽象库大多使用 C/C++写的，由此出现了 JNI，来完成 Java 对 C/C++库的调用；**

**安卓主许可证的 Apache，但整体上是由多个许可证混合组成的。**

**安卓代码组织与服务生态的描述**

**安卓内核使用的是经改造的 Linuxkernel，而其内核版本通常低于当前主流的 Linuxkernel 版本；**

**关于安卓系统定制与启动的观点安卓是有 uboot 引导的；**

关于编程语言框架

汇编语言编程专业难度最大，C 语言次之；

模型驱动编程语言更趋向于无代码编程；

嵌入式系统在不同层次，使用的语言可以是不同的。

关于嵌入式系统中 Java 编程及注释

Java 虚拟机就是一个虚拟化的微处理器；

关于 iOS 组织型编程语言的描述

遵循 HIG 保证了，苹果 App 的整体风格一致性；

关于嵌入式系统测试方面的观点

测试工具与测试 Benchmark 是测试的基础素材。

简答题：

一．1. 普适计算视角的专用计算，与通用计算相比特征体现在哪几个方面？

2. 普适计算内涵思考是什么？普适计算的颠覆性体现在哪几个方面？如何理解穿戴设备

是普适计算与边缘计算？

3. 嵌入式系统定义中，合理的代价是什么含意？如何看待嵌入对象的智能性、控制力和人机与群体交互三方面的分析？

1. 简述嵌入式系统的主要应用领域？说明云端组织计算的特征？

答：

1.普适性、透明性、动态性、自适应性、永恒性。

通用计算专于算力，专用计算则均衡算力与代价。而普适计算的基本思想是把计算机融入到环境中去，使人们关注的重点从操作工具转移到执行任务本身上来，可以在任意时间、使用任意设备、通过任意网络来获取所需的服务。

所以普适计算视角的专用计算，与通用计算相比，更加注重于算力与算力提高的代价间的均衡，而非专于算力的提高，而忽略由于算力提高而带来的代价。

2.基本思想：把计算机融入到环境中去，使人们关注的重点从操作工具转移到执行任务本身上来，可以在任意时间、使用任意设备、通过任意网络来获得所需的服务。

“消失”的技术：正如“书写”和“电”一样，由于这两种技术使用的普遍性导致它们变得非常平淡，以至忘记了这些技术对日常生活的巨大影响，犹如消失于人们的日常生活中。普适计算也将是那样一种消失的计算。

“不可见”的工具：一种好的工具是“不可见”的，它并不进入你的意识，你只是专注于任务而并非工具。普适计算也将是那样一种“不可见”的工具。

PC 所存在的一些根本问题（反过来说可以算普适计算的颠覆性吧）：

过分复杂而难以使用

过分要求人的注意力

过分隔绝于他人和现实活动

过分的支配作用使我们的桌面和生活犹如殖民地

3.（1）能耗代价、空间代价（体积大小、存在的环境）、成本以及由于性能和可靠性带来的代价。专用计算系统、定制化系统，重视对象和目标，与对象和目标配合以付出最合理的代价。

4.嵌入式系统有五大应用领域：国防电子、汽车电子、医疗电子、机床电子、消费电子。比如在无线传感网络、火炮控制、精确制导、电子侦察以及电视机、冰箱、微波炉等领域和产品中都大量的使用嵌入式系统。（嵌入式系统的主要应用领域包括工业、军事、通信、运输、金融、医疗、气象、农业等。）

1）基于虚拟化技术快速部署资源或获得服务；2）实现动态的、可伸缩的扩展；3）按需求提供资源、按使用量付费；4）通过互联网提供、面向海量信息处理；5）用户可以方便地参与；6）形态灵活，聚散自如；7）减少用户终端的处理负担；8）降低了用户对于IT专业知识的依赖。

二．1. 为什么说层次结构图适合构造系统，而对象结构图适合构造应用？

2. 控制数学模型与广义数学模型的异同处？

3. 如何理解嵌入式系统设计的三法则？

答：

1，层次结构图可以把一个大问题分解为几个层次的问题，简化解决问题的难度，所以层次结构图适合构造系统。

对象图可以明确设计需要，解决对象需求问题，更适合实现构造应用

2.同：都是为了简化要解决的问题的而构造的数学模型

异：控制数学模型主要是指描述控制系统及其各组成部分特性的微分方程、状态空间表达式、差分方程、传递函数、频率特性以及基于神经网络、模糊理论而建立的模型等。

广义数学模型则是用数学语言和运用数理逻辑方法建构的科学或工程模型。

3.洞察除法：本源追溯和交叉，强调简洁刚性度——功能与“系统和硬件”的完美匹配。如都江堰利用“鱼嘴分水”和“适中弯道”实现岷江的“分流”，以实现内流灌溉。

增强乘法：固本增强、群组协同，强调“自治互助度”——如都江堰内流利用“宝瓶口”控制灌溉水流，使其流量更适合于灌溉。

包容加法：自我排异、回滚包容，增强系统稳定度——如都江堰的“飞沙堰”，可以起到防洪排涝的作用。

三．1. 嵌入式微处理器指令集 5 种类型，说明其中 CISC&RISC，与进程&线程之间的关系，并

举出类似的范例？

2. 嵌入式系统硬件体系要素中的时钟种类，并说明其在系统群组能力中的基础意义？

3.如何理解 CMOS 工艺数字器件公式 P∝CV2F？进而说明嵌入式系统如何进行能耗优化控制的方法？锁相环从原理上如何用于调节系统功耗？

答：

1.指令系统从产生到现在的形态主要有五种风格；复杂指令集（CISC），精简指令集（RISC），超长指令字（VLIW），数据信号处理指令集（DSP），专用指令集（ASIP）。

CISC是一种为了便于编程和提高记忆体访问效率的晶片设计体系，常用指令只占20%，效率不高。

RISC是为了提高处理器运行速度而设计的晶片设计体系。关键技术在于流水线（pipelining）操作：在一个时钟周期里完成多条指令。提高了执行速度和可靠性，成本较低，但对编译器要求较高。

VLIW由编译器将多条可同时发送的指令并排在一条超长指令字中，并行执行多个操作，扩展了指令并行度。

DSP面向特殊应用，具有独特的体系结构，其芯片采用改进的哈佛结构，指令系统为流水线操作，数字信号处理可通过软件修改处理参数，灵活性强。具有良好的多机并行运行特性，采用专用硬件乘法器，电压较低。

ASIP的核心思想是，针对特定应用设计专用的指令集，再根据设计好的指令集优化处理器的体系机构，使优化后的体系结构更适合执行专用指令，进一步提高指令执行效率。

CISC：

–优点：指令越多功能越强，强调代码效率，容易和高级语言接轨。可以对存储器直接操作，实现从存储器到存储器的数据转移，可加入DSP指令。

–缺点：指令太多不易记忆；CPU内部结构复杂造成频率不高；指令执行速度慢。

RISC：

–优点：指令少容易记忆，尽量将操作码和操作数用1个16位数或32位数表示，指令整齐。CPU时钟频率可以做得很高，指令执行速度快。

–缺点：同样功能的程序，产生的代码量比较大；不能对存储器直接访问，不能实现存储器到存储器的数据转移。

2.时钟分类分为实时时钟（ RTC )、系统时钟、 CPU 时钟

RTC ：用于提供例如年月日时分秒和星期等实时时间信息，由后备电池供电，即使关闭系统，重启系统， RTC 仍然可以保持正确的时间和日期。

系统时钟：存储于系统内存中的逻辑时钟。用于系统的计算，比如超时产生的中断异常，超时计算就是由系统时钟计算的。这种时钟与 RTC 不同，在系统掉电或重新启动时会被清除。

CPU 时钟：即 CPU 的频率，时钟频率就是工作频率，即外频（主频＝外频×倍频）

3.P∝CV2F使用的前提是在数字集成电路中，CMOS电路静态功耗非常低，与动态功耗相比可忽略不计，故可以写成该形式。

公式中，P为CMOS芯片的动态功耗，C为CMOS芯片的负载电容，V为CMOS芯片的工作电压，F为CMOS芯片的工作频率。通过上式可发现，降低CMOS电路的工作电压比降低时钟频率更能降低活动功耗。因为功耗与电压时指数幂的关系，而与时钟频率只是线性关系。因此，选定满足电子系统中所有电驴模块的最低要求电压，对降低能耗至关重要。

第一 信号的调制和解调；第二 信号的调频和解调；第三信号频率合成电路。

四．1. ARM 公司发展最大启发是什么？芯片授权制造的最大难处是什么？

2. ARM 架构 Vs 处理器交织滚动发展的价值是什么？

3.ARM 处理器异常在计算架构上价值是什么？ARMv9 在 AI 指令增加上体现在哪些方面？

答：

ARM公司的最大启发是采用精简指令集架构相比于复杂指令集架构能够提供更高的性能和能效。

困难：核心技术过硬，融合包容性强

1、体积小、低功耗、低成本、高性能；

2、支持Thumb（16位）/ARM（32位）双指令集，能很好的兼容8位/16位器件；

3、大量使用寄存器，指令执行速度更快；

4、大多数数据操作都在寄存器中完成；

5、寻址方式灵活简单，执行效率高；

6、指令长度固定；流水线处理方式。

1可靠性和稳定性

2安全性

3错误排查与调试

ARMv9是ARM公司最新发布的架构版本，专注于提供强大的AI计算能力和安全性。在AI指令增加方面，ARMv9引入了以下特性SVE2，FP16ML，MVE，引入新的AI指令集扩展，提供了更强大的向量计算能力和优化的浮点数计算，

五．

1.简述 ARM 的常规存储设置映射？说明 little &big 多字节数据存储方式区别，并以 12H，

34H，ABH，CDH 为例分别示意存放顺序？

2.PC BIOS 和 ISA 总线在 PC 产业的发展过程中起到了哪些作用？为什么嵌入式系统中也加入

类 BIOS 的引导驱动层？

1. 简述 U-boot 两步启动过程？Stage1&Stage2, 按照本课程的讲解如何理解最小系统的三分类？

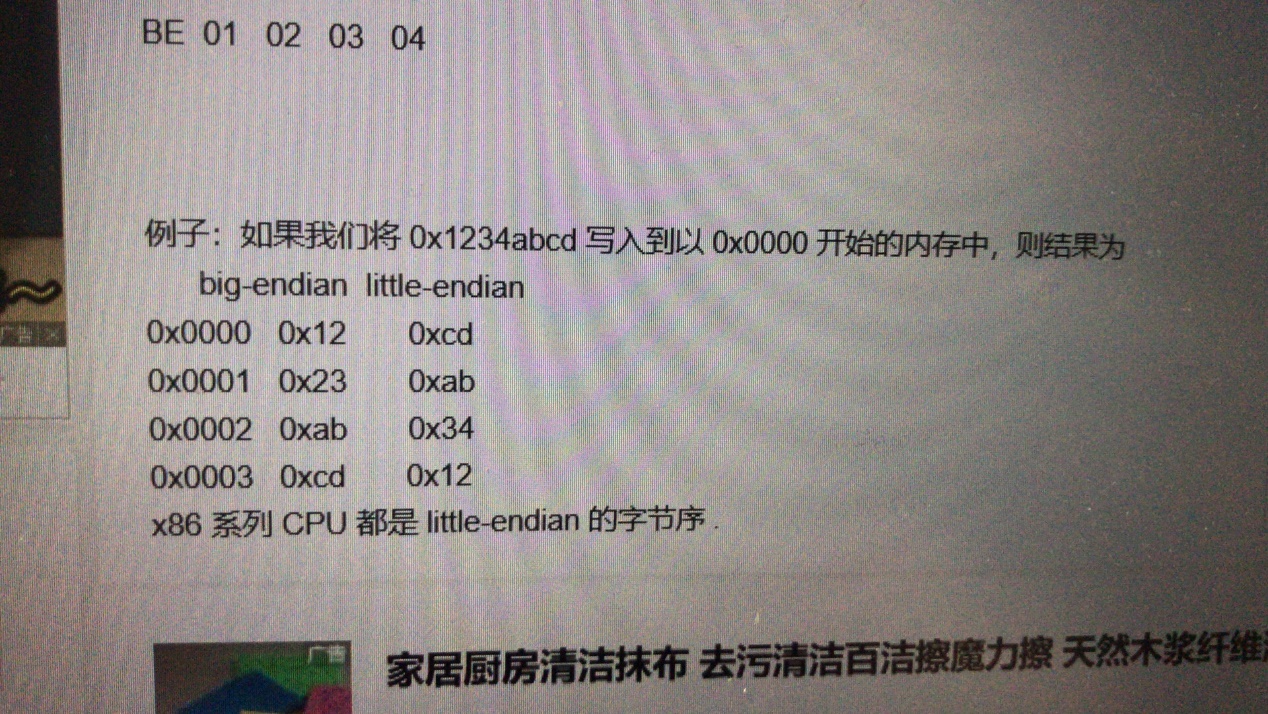
答：

1.big endian是指低地址存放最高有效字节（MSB），而little endian则是低地址存放最低有效字节（LSB）。

12H：0x00010010

Big endian:00 01 00 10

Little endian:10 00 01 00



2.PC结构中的BIOS的出现产生了具有重大意义的变革，它促使了硬件与软件就得以分离，使得PC软件得以独立发展，而且由于有了BIOS，增加了I/O扩展槽，才有了外部总线，促使I/O设备得以应用。

ISA总线使寻址范围、数据总线宽度、中断处理能力有了很大的提高，具备了良好的兼容性，为以后更先进总线的诞生打好了基础。

BIOS在嵌入式系统中具有极其重要的作用，主要表现在一下三方面：

1. 自检及初始化。这部分工作是负责启动计算机，主要完成三件事：对硬件进行检测；创建中断向量、设置寄存器、对一些外部设备进行初始化和检测、设置BIOS的参数；引导操作系统。

2. 程序服务处理，这些工作主要是为应用程序和操作系统服务的。服务内容主要与I/O设备有关，例如读磁盘，文件输出到打印机等。

3. 中断处理，在开机时，BIOS会传送所有硬件设备的中断向量号给CPU。当用户发出使用某个设备的指令后，CPU就会根据中断向量号使用相应的硬件来完成工作。

3.（1）第一阶段的功能

Ø  硬件设备初始化

Ø  加载U-Boot第二阶段代码到RAM空间

Ø  设置好栈

Ø  跳转到第二阶段代码入口

（2）第二阶段的功能

Ø  初始化本阶段使用的硬件设备

Ø  检测系统内存映射

Ø  将内核从Flash读取到RAM中

Ø  为内核设置启动参数Ø  调用内核

六．1.解释微内核与层次内核 OS 异同处？并说明组织型系统优势？

2.说出自由/开源软件许可证主要种类与 LinuxKernel 版本质量种类？

3.怎样评测嵌入式操作系统的实时性？描述评价嵌入式操作系统的 10 大方面？

答：3.实时性能指标包括任务切换时间、中断延迟时间和系统响应时间。

任务切换时间：该指标主要受微处理器控制，在进行任务切换时，保存和恢复CPU所花费的时间与CPU的寄存器个数有关。

中断延迟时间：由最大中断屏蔽时间，得到只想ISR（中断服务例程）向量的时间、存储所有寄存器的时间和执行中断处理的时间。

系统响应时间：该时间是系统在发出处理要求到系统给出应答信号的时间，从整体上评价操作系统。

实时性测量方法：Rhealston方法、进程调度间隙时间法、三维表示法、Real/Stong Benchmark方法。

方法说明：

Rhealstone方法（六元素测量法）：对EOS的六个关键操作的时间量进行评测，即任务切换时间，抢占时间，中断延迟时间，信号量混洗时间，死锁解除时间、数据包吞吐率。

进程调度间隙时间法：从中断产生到由中断激活的实时任务开始执行之间的时间间隔。

三维表示法：测试EOS的计算能力、中断处理能力、I/O吞吐率。然后可用一个三维图形来表达三个特性之间的依赖关系。

Real/Stong Benchmark方法：为一个纯软件的基准程序，属于合成工作负载，包含了系统响应能力、系统抢占能力和系统I/O吞吐能力三个测试。

十个方面分别是时域、资源占用、能耗、干扰、智能、控制、融合、封装、功能性、可靠性。

时域：系统完成规定任务所需要的时间必须满足的条件。

资源占用：系统在正常运行的各个阶段，所需的硬件资源的种类，数量及多少。

能耗：系统在运行的各种状态下所消耗的电能的速率。

干扰：嵌入式产品在加入了嵌入式系统后，对原有产品性能及其他指标的影响程度。

智能：统能够完成的操作种类多少、解决问题的复杂程度以及自我完善、更新、发展的能力。

控制：系统的控制过程的快速、准确和精细程度以及操作此控制作用的便捷性。

融合：指嵌入式系统和嵌入式产品其他部分相匹配、结合的能力。

封装：简单说，封装就是打包，把一系列相关的东西打包在一起，把细节隐藏，只把简单的、稳定的接口提供给用户。

功能性：拥有多种功能

可靠性：在预定的操作和环境条件下，在一定的时期或是某一时刻，系统专一并正确的执行任务的可依赖程度。

七．1.AndroidOS 是第一大 OS？但其是传统意义下的通用 OS 吗？

1. 如何理解安卓多子松组织生态架构？DVM 的开放价值？
2. 安卓系统 NDK 开发与应用 SDK 开发区别？JNI 有何作用？

八．1.如何理解 Java 体系及其程序兼容性，JVM 起到了什么作用？

2.怎样理解 NUI 及基于 iOS 组织型编程？HIG 有关 UI 方面的 5 项准则分别是什么？

3.软件测试基本机理是什么？复杂的软件系统能够保证绝对没有 Bug 吗？系统设计师永远知道自己设计的系统死穴是上面意思？

答：1.Java语言的安全性可以从两个方面得到保证。

一方面在Java语言里，象指针和释放内存等C++功能被删除，避免了非法内存操作。

另一方面当Java用来创建浏览器时，语言功能和一些浏览器本身提供的功能结合起来，使它更安全。

Java语言在机器上面执行前，要经过很多次的测试。它经过代码校验、检查代码段的格式、检测指针操作、对象操作是否过分以及试图改变一个对象的类型。

虚拟机是由一套完整的规范定义的。JVM保证了Java代码在任何系统上到能运行。凡是符合Java虚拟机规范的实现都是百分之百兼容的。JVM规范对其实现做出了具体规定。JVM规范定义了虚拟机体系结构、支持的数据类型和类结构文件。

2.（1）一致性原则

坚持以用户体验为中心设计原则，界面直观、简洁，操作方便快捷，用户接触软件后对界面上对应的功能一目了然、不需要太多培训就可以方便使用本应用系统。

（2）准确性原则

使用一致的标记、标准缩写和颜色，显示信息的含义应该非常明确，用户不必再参考其它信息源。

（3）布局合理化原则

在进行UI设计时需要充分考虑布局的合理化问题，遵循用户从上而下，自左向右浏览、操作习惯，避免常用业务功能按键排列过于分散，以造成用户鼠标移动距离过长的弊端。多做“减法”运算，将不常用的功能区块隐藏，以保持界面的简洁，使用户专注于主要业务操作流程，有利于提高软件的易用性及可用性。

（4）系统操作合理性原则

尽量确保用户在不使用鼠标(只使用键盘)的情况下也可以流畅地完成一些常用的业务操作，各控件间可以通过Tab键进行切换，并将可编辑的文本全选处理。

（5）系统响应时间原则

系统响应时间应该适中，响应时间过长，用户就会感到不安和沮丧，而响应时间过快也会影响到用户的操作节奏，并可能导致错误。

3.1)机理：软件测试是使用人工或自动的手段测试某个软件系统的过程。目的在于检验它是否满足规定的要求或弄清预期结果与实际内容之间的差别。

（软件测试的方法包括白盒测试和黑盒测试。白盒测试：按照程序内部的结构测试程序，通过测试来检测产品内部动作是否按照设计规格说明书的规定正常运行，检测程序中的每条通路是否都按照预定要求工作。黑盒测试：也称功能测试，是通过测试来检测每个功能是否都能正常使用，完全不考虑程序内部结构和内部特性。）

2)不能保证。测试程序是不能发现所有Bug的，只能将软件系统中可能存在的Bug找出来，但是并不是所有的Bug都能被发现。但没有致命的BUG存在，模拟的不完整。

3)

九．

1. 系统失效、错误与缺陷,系统缺陷静止、活跃与激活,相互关系是什么？

2. 嵌入式系统的可靠性主要是指哪些属性？几种提高嵌入式系统可靠性的手段各有什么特

点，如何选取合适的手段用于系统设计？

3. 说明什么叫系统缺陷？以下面代码片段为例，软件缺陷发生与状态激活的变化？如何抑

制该缺陷，并做简单说明？

如下代码片段：

1:float func(fload f1, float f2)

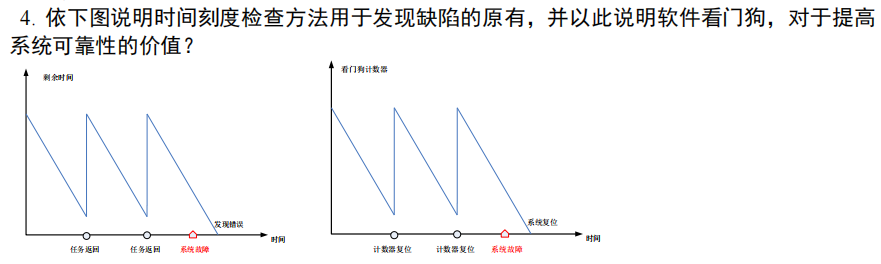
2:{

3: float result;

4: result = 1/(f1-f2);

5: return result;

6:}



答：

2.（1）可用性：随时可以向使用者提供正确服务。

可信性：可以持续的提供正确服务。

安全性：不会对使用者和环境造成灾难性的损害。

完整性：不会出现意外的系统部分替换。

可维护性：可以进行修改和维修。

（2）如何正确合理的处理缺陷，是提高嵌入式系统可靠性的关键。

处理缺陷的方法：

1.缺陷避免：避免缺陷出现或被引入系统。

2.缺陷包容：在缺陷出现时，避免出现系统服务失效。包括错误发现和恢复。

3.缺陷取出：减少缺陷的数量，降低失效的严重性。

4.缺陷预测：预测缺陷的数量，分布，可能产生的结果等。

3.a,确保每个被发现的缺陷都能够被解决；

b,这里解决的意思不一定是被修正，也可能是其他处理方式（例如，在下一个版本中修正或是不修正），总之，对每个被发现的BUG的处理方式必须能够在开发组织中达到一致；

c,收集缺陷数据并根据缺陷趋势曲线识别测试过程的阶段；决定测试过程是否结束有很多种方式，通过缺陷趋势曲线来确定测试过程是否结束是常用并且较为有效的一种方式；

d,收集缺陷数据并在其上进行数据分析，作为组织的过程财富。

4.嵌入式系统中软件看门狗实现多线程监测以及运用/proc文件系统实现监测数据采集的原理和方法，讨论了使用UML对自监测系统进行面向对象的分析以及设计，并将该方案应用到TI公司的开发板实验。测试结果表明，它可以有效的实现对被监测进程的每一个线程的监测，并且可以对系统的平均负载、内存剩余量和指定接口收发包 情况进行监测，确保系统的长时间的运行。

十．1. 怎么理解对象优化与系统优化？嵌入式系统优化分为哪三个方面？

1. 如何理解代码注释优化与能耗指标优化？

答：1.（1）等效性原则

优化前后的代码实现的功能要完全一致

（2）有效性原则

优化后的代码一定要比原先的代码运行速度快活着占用存储空间小，或者二者兼有，否则就是毫无意义的优化

（3）经济性原则

很多代码性能低下的部分原因也是由于硬件性能的限制，比如将文件压缩存放以节约存储成本。优化要在现有的条件下考虑，不要以更换存储空间的大小来换取解压的时间。优化要付出较小的代价，很多程序员在做优化的时候，抱怨设备的性能有限，要求提高设备的性能，这样只能是本末倒置。

十一．1.如何理解 MiniSystem、SamllSystem 与 StandardSystem 系统分类？

2.嵌入式系统的数据消创是指什么？举例说明。

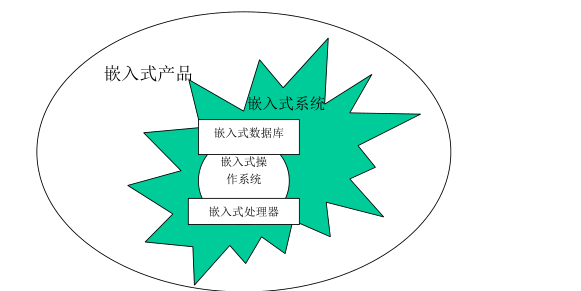
3.嵌入式系统的多周期性，如何理解？

图论：

**1． 园环套图概念关系：嵌入式处理器/嵌入式操作系统/嵌入式数据库/嵌入式系**

**统/嵌入式产品相互关系**

嵌入式微处理器：指具备强的中断、I/O、内存和能耗管理能力、具有定制多类体系构架的特征，适当计算处理性能的微处理器



关系：嵌入式产品包含了嵌入式系统，但并不是嵌入式系统。嵌入式系统包含嵌入式数据库、嵌入式操作系统、嵌入式处理器，其中嵌入式操作系统是嵌入式系统的核心，嵌入式处理器是硬件基础，嵌入式数据库是嵌入式系统的上层应用。

2.

是嵌入式系统的五层结构模型。五层模型最底层为硬件层，这是系统架构的基础，所有的功能实现与应用都是以此为基础的。硬件层以上是引导层，它是静止的硬件与动态运行的嵌入式系统之间的过渡。硬件层与引导层一起可以构成一个最小系统。引导层之上是内核层，实质上就是指嵌入式操作系统。第四层是SUI是计算机和使用者之间的对话接口。最上层为用户层，是面向用户的，大多数程序都在这一层进行工作。其中，内核层与SUI这两层又统称为中间件。内核层实际上就是指操作系统，UI层为图形用户界面接口（GUI）及其他系统与用户接口。

1. 看问题的角度不同。五层模型综合了系统的物理与逻辑架构，兼顾了系统宏观与微观的层次要素，体现了系统的思想，是由内到外的。对象化仅仅是逻辑上的，反映了系统的应用角度和内在形态，在逻辑上是完全的，也是系统的。但对于五层结构来说，它反映的是五层结构的逻辑应用的一个侧面，是由外向内的。

2. 层次间关系不同。嵌入式五层模型是一个层次分明、结构严谨的框架，且具有自己的约束条件，各层顺序严格规定，每一层只与相邻的层发生关系，比环形结构更加系统。对象化环是一个相对松散的结构，各个模块间是并列的关系，符合逻辑应用的侧重点。

3. 侧重点、功能不同。五层模型侧重系统整体的设计与分析，对象化环模型则侧重于逻辑应用。

该图是嵌入式系统的五层裁剪图，用以对嵌入式系统进行抽象、概括和描述，方便对系统进行实验、分析与预测，并且作为一个框架，用来协调各层标准的制定。嵌入式系统的五层结构分别是应用层、UI层、内核层、引导驱动层和硬件层。

最底层的硬件层包括供电、时钟、处理器、总线，还有内存、键盘、显示、接口部分。引导驱动层下面是引导及固件，上面则由驱动作为内核的接口。硬件和引导驱动层共同构成了最小系统。

再上一层是内核层，包括相对底层的OS内核部分，它连接着内核与驱动。也包括文件、网络、任务等。UI层为使用者界面，包括SUI、文件、网络、任务。而应用层则是与消费者最接近的一层，是由各种应用程序需组成。

3.

上图从应用角度给出嵌入式微处理器的定义：指具备强中断、I/O、内存和能耗管理能力，具有定制多类体系架构特征，适当计算处理性能的微处理器。而这些功能、特性由中间的指令系统支持，微处理器与指令系统密不可分。

指令系统从产生到现在的形态主要有五种风格；复杂指令集（CISC），精简指令集（RISC），超长指令字（VLIW），数据信号处理指令集（DSP），专用指令集（ASIP）。

CISC是一种为了便于编程和提高记忆体访问效率的晶片设计体系，常用指令只占20%，效率不高。

RISC是为了提高处理器运行速度而设计的晶片设计体系。关键技术在于流水线（pipelining）操作：在一个时钟周期里完成多条指令。提高了执行速度和可靠性，成本较低，但对编译器要求较高。

VLIW由编译器将多条可同时发送的指令并排在一条超长指令字中，并行执行多个操作，扩展了指令并行度。

DSP面向特殊应用，具有独特的体系结构，其芯片采用改进的哈佛结构，指令系统为流水线操作，数字信号处理可通过软件修改处理参数，灵活性强。具有良好的多机并行运行特性，采用专用硬件乘法器，电压较低。

ASIP的核心思想是，针对特定应用设计专用的指令集，再根据设计好的指令集优化处理器的体系机构，使优化后的体系结构更适合执行专用指令，进一步提高指令执行效率。

4.

启动前操作系统仅仅是存储在不挥发介质中的代码，从启动的一刻开始，操作系统将被装载到内存中，成为完整功能的运行系统。这个装载的过程就是引导。引导是系统启动和运转的第一步，是系统从硬件走向系统软件的开始，从这一刻开始，系统将从固化的代码和静态的电路转化为动态运行的程序和具有功能的系统。2.为了保证系统功能可以正常调用，所做的初始化准备就是驱动。3.顺利进行引导与驱动两部分工作就是引导层要做的工作。

融合度是指引导层对于下级的硬件层控制和兼容能力。由于硬件的参差不齐，所以融合度不同，才有了引导与驱动这一层，厚度是指引导层的规模。一般来说，引导层的不稳定性与引导层的规模成正比。平整度是指引导层对上级的中间件层所提供的操作接口的统一性和友好性。

引导层的作用之一就是屏蔽掉底层硬件的差异性，为上层的操作系统和各种程序调用提供统一的接口。如果引导层任务太多，虽然会向上提供一个平整的接口，但效率会明显下降。如果引导层太薄，效率会提高，但设备无关性会降低，应结合具体应用分析。引导层的平整度、厚度和融合度是衡量引导层的重要指标。

最小系统分两种，一是OS内核，引导和驱动，而是引导和驱动。它是一个含有微处理器的系统。具有制造功能和用户功能所需的最少硬件及软件环境，是嵌入式系统能工作的最低要求，不具有智能性，是系统成长的基础。

5.

整体内核（单内核，大内核），是一种几乎涵盖了所有操作系统功能模块，如进程调度、进程间通信、内存管理、设备管理、文件系统、网络系统的内核结构。大部分模块遵守特定的接口规范，相互协调。所有模块在编译时链接在一起，形成一个可执行文件。运行时所有操作系统功能模块处于内核态，而其他用户程序和系统程序处于用户态，通过系统调用可以切换到内核态。但调试困难，难于裁剪，稳定性相对较差。如WinCE就是采用此结构。

插件式结构（微内核结构），是一种类似于C/S模式的内核结构。微内核运行在核心态，提供所有操作系统的基本操作。结构规范，易于裁剪和编程，且调试简单。扩展性和移植性、鲁棒性好，但整体效率低，资源管理复杂。如QNX就是采用此结构。

层次内核结构是一种基于分层思想的内核结构。它把操作系统的所有功能划分为若干模块，按功能流程的调用次序排列成若干层。（下层模块封装内部细节并向上提供统一的抽象和接口，上层模块调用下层模块提供的接口，各个层级间的实现比较独立。各个模块间的组织结构和依赖关系清晰明确，适合进行系统功能的划分）分层思想有利于组织操作系统的开发，易于调试和诊断，系统的修改和扩展相对容易。但是对系统功能进行定制和裁剪相对困难，且它是一个理论化的结构，实际中是没有完全符合层次的操作系统。

6.

嵌入式系统的五层结构中与用户最近的一层就是第五层应用层。而应用层则是由各种程序组成的，这些程序又出自于各种不同的程序设计语言，比如Java、C+、C、汇编语言等。该图把程序及语言放在五层结构的上面，意思是对于普通的用户和消费者而言，这些才是看得到用得到的部分，它们遮盖住了嵌入式系统的复杂结构，是嵌入式系统与用户的统一接口。任何系统的根本目的都是为了对数据进行处理，而这种处理正是通过程序这种特殊的数据来实现的。如何保证应用程序及数据的开放和兼容性，是行业面临的基础问题。因此最理想的情况下是软件及程序能够具有非常好的兼容和移植性，能够使用户看不出复杂的底层结构上的差别。

**8.**

该图画出了嵌入式系统的测试方法：功能测试、极限测试、崩溃测试、异常测试，以及可靠性、裁剪性、兼容性、安全性测试。

功能测试时嵌入式系统测试的最基本本分，同时崩溃测试、极限测试、异常测试也是非常重要的测试环节。而且可靠性测试可以测试系统的可靠程度，裁剪性测试则测试出了系统的裁剪性，兼容性测试可看出系统的兼容程度，而异常测试则是测试系统是否会发生异常。只有这样才能尽量使系统具有最好的稳定性。

**9.**

此图为设计结构的八角图，也称为蜘蛛图。嵌入式系统的特性可以分为可信系统与非可信系统，可交换系统与非可交换系统，实时系统与非实时系统，可信系统与非可信系统四种。任何一个操作系统都不可能同时满足任何对角的系统特性，这就必须要根据实际的需求来选择各个属性的程度，来组建一个最符合用户需求的嵌入式系统。在实际的工作中，往往也需要靠经验和对规律的总结来设定到底采用哪些系统方式。

**10.**

该图类似于太极阴阳图，是对于整个嵌入式系统的总括。中间的硬件和软件是两个不同的部分，但又通过某些方式交织在一起，互相对立，又互相包含。外层则是围绕着整个嵌入式系统的软硬件的一种动态的流程与方法，从创建到设计，再由设计到制造，由制造到维护，由维护又回到创建。如此循环往复，周而复始，使得整个系统动静结合，得以不断发展完善。