

## 【第五章·章节问题】相关答案请浏览 NK2024 春·embed-lecture7(ch5+无测试点).pptx、NK2024 春·embed-lecture9(ch5microkerne+无测试点 I).pptx 文件及视频回放

### 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？

(我自己根据PPT编的)

**同：**都把内核分为了内核级与用户级，都含有操作系统的核心功能和基本功能。

**异：**微内核只含有操作系统的核心功能和基本功能，微内核将传统操作系统的一些功能从内核中分出，以子系统形式相互作用。微内核结构结构规范,易于裁剪、编程和调试，扩展和移植性强，降低内存固定使用量，实时性强；但整体效率低，资源管理复杂。层次内核体系结构各模块之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、调试和诊断，也使得系统的修改和扩展相对容易。但对某一层功能的修改往往影响到相关层次,使得指定和裁剪系统功能相对困难;底层代码和硬件直接相关；功能越强,关系越紧，与微内核结构相比比较庞大，移植较复杂。

#### 组织型系统的优势：

- 把一个复杂的操作系统分解成许许多多功能单一的模块。各模块之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，易于测试和诊断，也使得系统的修改和扩展相对容易。

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

(答案来源：lecture7 PPT第24页+76页，版本质量种类我抄以前的答案，我没找着这部分，他在版本那部分也没介绍质量种类)

**源软件许可证主要种类：** GPL、LGPL、MPL、MIT、BSD、Apache

Linux Kernel主要质量版本：

mainline: 5.6

stable: 5.6.3

stable: 5.5.16

longterm:5.4.31

longterm:4.19.114

longterm:4.14.175

longterm:4.9.218

longterm:4.4.218

longterm:3.16.82

linux-next:next-20200412

### 3.怎样评测嵌入式操作系统的实时性？

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。

#### 4.描述评价嵌入式操作系统的 10 大方面？

10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：系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
- 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。

### 【OnAndroid专题3问】相关答案请浏览NK2024春·embed-lecture7-1(ch5+无测试点).pptx文件及视频回放

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

AndroidOS 是目前全球市场份额最大的移动操作系统，但其主要设计和优化方向是移动设备，而非传统意义上的通用操作系统。尽管其具备一定的跨平台能力和丰富的应用生态系统，但在桌面和服务器应用支持、用户界面设计和开发环境等方面仍与传统的通用操作系统（如Windows、Linux、macOS）存在差异。因此，AndroidOS 更适合作为一种移动优先的操作系统，而不是传统意义上的通用操作系统。

#### 2. 如何理解多子松组织生态架构？DVM 的开放价值？

多子松组织生态架构为多子生态与松组织的结合，多子生态即说明系统由更多的子系统组成，松组织则说明系统内部有更丰富的组成，既有开源，又有闭源。

**DVM的开放价值：**（Dalvik是Google公司自己设计用于Android平台的Java虚拟机。）它可以支持已转换为.dex（即Dalvik Executable）格式的Java应用程序的运行，.dex格式是专为Dalvik设计的一种压缩格式，可以减少整体文件尺寸，提高I/O操作的类查找速度，适合内存和处理器速度有限的系统。DVM也是基于寄存器的虚拟机(Register-based)，可以根据硬件实现更大的优化，适合在移动端使用。

### 3. 安卓系统开发与应用开发区别？JNI 有何作用？

**区别：**

#### **(1) Android系统开发（底层开发）**

Android系统开发（移植）属于底层的开发工作，主要针对Android系统本身进行完善和将其移植到其它的硬件平台，因此需要掌握Android系统构架中的Linux内核层和系统运行库层。

Android系统开发主要涉及Libraries和Android runtime这两部分内容，一般就是在本地编写C/C++代码，然后通过JNI向上层提供调用接口，主要是为应用开发提供服务等工作。

#### **(2) Android应用开发（上层开发）**

Android应用开发主要是根据Android系统所提供的API来完成各种应用程序的编写，所使用的API属于Android的应用框架层。如果Android系统没有提供某些API，那么只能通过Android系统底层编写C/C++代码来开发这些API并向Android应用框架层提供接口，但是这样有可能会使应用程序与其他Android系统不兼容。应用开发属于上层开发。

**JNI的作用：**

提供了若干的API实现了Java和其他语言的通信（主要是C&C++）。

从JVM角度，存在两种类型的代码：“Java”和“native”，native一般指的是c/c++，为了使java和native端能够进行交互，java设计了JNI（java native interface）。JNI允许java虚拟机（VM）内运行的java代码与C++、C++和汇编等其他编程语言编写的应用程序和库进行互操作。

## **【第六章章节问题】 相关答案请浏览NK2024春·embed-lecture11(ch6+无测试点).pptx文件及视频回放**

### **1. 如何理解 Java 语言的安全性、程序兼容性，JVM 起到了什么作用？**

#### **(1) Java 语言的安全性**

Java语言删除了类C语言的指针和内存释放等语法，有效地避免了用户对内存的非法操作

Java程序代码要经过代码程序校验、指针校验等很多测试步骤才能够运行。所以未经允许的Java程序不能出现损害系统平台的行为。

Java可以编写出防病毒和防修改系统。

#### **(2) 程序兼容性（体系结构中立）**

因为Java编译器编译生成了与体系结构无关的字节码结构文件格式，所以使用Java语言编写的查询，只要做较少的修改，甚至有时根本无须修改就可以运行在所有的平台上。

体系结构的中立是指Java编译器通过伪编译后，生成一个与任何计算机系统无关的“中性”的字节码，这种字节码经过了进行的设计，能够很好地兼容于当今大多数流行的计算机系统，并且体系结构的中立也使得Java系统具有可移植性。Java应用程序可以移植到不同的处理器和操作系统上进行编译和运行。

## JVM的作用

JVM是运行Java字节码的虚拟机。JVM有针对不同系统的特定实现（Windows, Linux, macOS），目的是使用相同的字节码，其均会给出相同的结果。字节码和不同系统的JVM实现是Java语言“一次编译，随处可以运行”的关键所在。

### 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？

- (1) 隐喻：尽可能按照现实世界中的对象和操作来对您程序中的对象和操作建模。
- (2) 直观操作：直观操作意为人们感到他们在物理上，而非抽象地控制某物。
- (3) 即看即点：让用户做出选择，而不是要求他们进行开放式的文本输入
- (4) 反馈：您的程序应通过一些视觉上的变化来响应用户的每一个操作
- (5) 用户控制：让用户来触发和控制操作，而不是您的应用程序。
- (6) 美学完整性：美学上的完整性并不是用来衡量您的程序有多漂亮，而是衡量您程序的外观和程序功能结合得有多好。

### 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？

- (1) 软件测试的基本机理：分类：崩溃测试、用户测试、极限测试、异常测试
- (2) 复杂的软件系统不能够保证绝对没有 Bug。
- (3) 系统设计师虽然对系统非常熟悉，但由于各种限制，也不能完全预见所有的“死穴”。通过不断的测试、监控和改进，可以尽量减少软件中的错误和缺陷，提高系统的可靠性和稳定性。

## 【系统可靠性专题问题】

**相关答案请浏览NK2024春·embed-lecture12(Dependability&performance).pptx文件及视频回放**

### 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？

#### (1) 系统失效、错误与缺陷

失效(failure)：从系统提供正确的服务向系统提供错误服务的状态迁移的过程。

错误(error)：系统失效时，系统提供的服务与正确的服务之间的偏差。

缺陷(fault)：直接或间接引起错误的原因。

相互关系

#### (2) 系统缺陷静止、活跃与激活

缺陷活跃(active)：缺陷引发错误

缺陷静止(dormant): 缺陷存在但不引发系统错误

缺陷激活(activation): 利用一个外部输入将静止态的错误转变为活跃态的过程。

相互关系: 活跃、静止、激活是缺陷的三种不同的状态。

## 2. 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, 并做简单行间注释与段注释?

如下代码片段:

```
float func(float f1, float f2)
{
    float result;

    result = 1/(f1-f2);

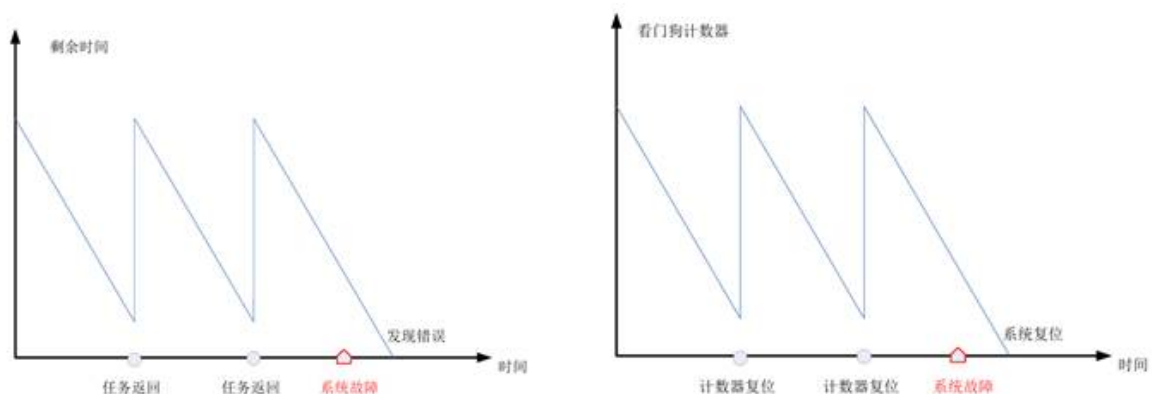
    return result;
}
```

此函数中, 第4行若出现 $f1=f2$ 的情况, 就会出现除0异常, 引发错误, 因此, 第4行语句中存在缺陷。

(1) 对于 $\text{func}(3.0, 5.0)$ ,  $\text{func}(100.0, -0.5)$ 等调用, 程序可以正常运转, 并得出正确的结果, 此时称为缺陷静止(dormant)

(2) 当 $\text{func}(4.0, 4.0)$ 这类调用发生时, 错误被输入激活(activation); 此时程序运行会引发除0错中断, 系统运行被打断, 函数执行错误, 功能失效, 缺陷处于活跃(active)状态

## 3. 依下图说明时间刻度检查方法用于发现缺陷的原有, 并以此说明软件看门狗, 对于提高系统可靠性的价值?



两张图反映时间刻度检查

### (1) 图一

每一个任务的执行或每个代码段的执行都需要在限定的时间内完成

对于实时系统, 每一个任务的执行和返回时间都需要有严格的限制

如果程序运行超出指定的时间, 则说明程序进入失控状态, 此时系统已经发生错误

## (2) 图二

看门狗计数器用于保证系统的可靠性

利用周期性运行的系统任务复位看门狗定时器。如专门的看门狗进程，或系统的调度器，周期性的向定时器的计数器中写入计数值

看门狗电路中自动将计数器的值递减。当此值递减为0时，触发系统复位

## 【系统优化专题】

**相关答案请浏览NK2024春·embed-lecture12(Dependability&performance).pptx文件及视频回放**

### 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？

(编的，这里太抽象了)

**对象目标优化：**关注具体、可量化的目标，通过针对性的措施来实现这些目标的优化。

**系统优化：**通过对系统内部各个部分进行协调和优化，实现整体系统的效率最大化和长远影响。

**优化三个方面：**

- 运行速度
- 优化代码尺寸优化
- 低功耗优化

### 2. 如何理解代码注释优化与能耗指标优化？

(编的，没有这玩意)

代码注释优化和能耗指标优化都是提高系统整体质量和效率的重要手段。代码注释优化侧重于提高代码的可读性和可维护性，而能耗指标优化则侧重于降低系统运行时的能源消耗。两者的目标都是通过优化过程中的某些方面，提升系统的性能和用户体验。在嵌入式系统课程设计中，这些优化措施被明确量化并作为验收标准，确保学生在实践中掌握优化技能，最终实现高质量的项目成果。

**【第七章章节问题】相关答案请浏览NK2024春·embed-lecture13(ch7)文件及视频回放**

### 1. 如何理解 MiniSystem 、 SamIISystem 与 StandardSystem 系统分类？

- 轻量系统 (MiniSystem)：面向MCU类处理器例如Arm Cortex-M、RISC-V 32位的设备，硬件资源极其有限，支持的设备最小内存为128KiB，可以提供多种轻量级网络协议，轻量级的图形框架，以及丰富的IOT总线读写部件等。可支撑的产品如智能家居领域的连接类模组、传感器设备、穿戴类设备等。
- 小型系统 (SamIISystem)：面向应用处理器例如Arm Cortex-A的设备，支持的设备最小内存为1MiB，可以提供更高的安全能力、标准的图形框架、视频编解码的多媒体能力。可支撑的产品如智能家居领域的IP Camera、电子猫眼、路由器以及智慧出行域的行车记录仪等。

- 标准系统 (StandardSystem)：面向应用处理器例如Arm Cortex-A的设备，支持的设备最小内存为128MiB，可以提供增强的交互能力、3D GPU以及硬件合成能力、更多控件以及动效更丰富的图形能力、完整的应用框架。可支撑的产品如高端的冰箱显示屏。

## 2. 嵌入式系统的数据消创是指什么？举例说明。

**数据消创：**数据的迁移使用，指在物联节点既消费数据，又创造数据与他人分享。

**举例：**在纸联当中某个用户创造的好的笔记（数据创造），可能可以给其他用户提供帮助和参考（数据消费）。

## 3. 嵌入式系统的多周期性，如何理解？【我回看了课老师上课没讲我乱诌的SOS】

从结构的角度看，嵌入式系统对实时多任务有很强的支持能力，能完成多任务并且有较短的中断响应时间，具有多周期性。

从功能的角度看，以体联·五时健康为例，涉及体力、情绪、智力的多周期节律，在功能上支持多周期性。