

第一章章节问题

1. 计算载体与计算系统，与代价关系简述？通用计算与普适专用计算的区别？

<关系简述>

计算系统包含计算载体。计算系统还包括在计算载体上运行的程序、数据，以及各部分和部分之间组织起来的设计思想等。计算载体只是承载思想和数据运行的实体而已。

它们之间的代价关系可以从几个方面来简述：

1、硬件成本：计算载体的选择直接影响了成本。购买更高性能的服务器或云服务器可能需要更高的投资，而选择性能较低的个人电脑则成本较低。因此，在考虑代价时，需要权衡硬件的性能和成本。

2、软件成本：计算系统所需的软件许可费用也是一个重要考虑因素。有些软件需要购买许可证，而有些则是开源免费的。此外，一些企业可能需要定制开发软件，这也会增加软件开发和维护的成本。

3、运维成本：随着计算载体和计算系统的规模增加，运维成本也会相应增加。这包括硬件维护、软件更新、安全管理等方面的成本。例如，使用云服务器可能会降低硬件维护的成本，但需要支付额外的服务费用。

4 性能与效率：计算载体的性能和计算系统的效率直接影响到工作负载的处理速度和效果。更、高性能的计算载体和优化的计算系统可以提高工作效率，从而降低时间成本和人力成本。

5、扩展与升级成本：随着业务需求的增长，可能需要扩展计算载体和升级计算系统。这涉及到硬件的扩容和软件的更新，也会增加相应的成本。因此，在选择计算载体和计算系统时，需要考虑其扩展性和升级路径，以减少未来的成本开销。

<通用计算与普适专用计算的区别>

(1) 适应性上，专用计算适应性差，不适用于除特定针对某类问题以外的其他方面，而通用计算应用面广，适应性强；

(2) 效率上，专用计算能在其擅长的特定的方面展现出最有效、最快速和最经济的特性，而通用计算的运行效率、速度和经济性依据不同的应用对象会受到不同程度的影响；

(3) 功能上，专用计算是为适应某种特殊需要而设计的，增强了某些特定功能，忽略了一些次要要求，而通用计算广泛适用于一般科学运算、学术研究、工程设计上，具有功能多、配置全、用途广等特点。

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？

<内涵思考>

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。

<颠覆性>

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）：

1. 过分复杂而难以使用
2. 过分要求人的注意力
3. 过分隔绝于他人和现实活动
4. 过分的支配作用使我们的桌面和生活犹如殖民地

<穿戴设备是普适计算与边缘计算>

普适计算是一种人们能够在任何时间、任何地点、以任何方式进行信息的获取与处理的计算；是强调和环境融为一体的计算；在普适计算模式下，计算机本身则从人们的视线中消失。既然普适计算强调以人为中心，计算的载体在当前阶段就落在了“物”——用户不觉察的情况下进行计算、通信，提供各种服务的“物”，可穿戴设备就可以“随时随地”提供这样轻松自然并且可以不被用户察觉到的服务。

边缘计算是为应用开发者和服务提供商在网络的边缘侧提供云服务和 IT 环境服务；目标是在靠近数据输入或用户的地方提供计算、存储和网络带宽。其低延迟、低带宽运行以及保护隐私的优点与穿戴设备的设计所需高度一致。

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？

<合理的代价>

以合理的代价提高对象体系智能性、控制力和人机与群体交互能力为目的，通过相互作用和内在指标评价的，嵌入到对象体系中的专用计算机系统。其中合理的代价主要包括对嵌

入式系统前期的开发问题、中期的资源如何利用，又包括成本以及实现嵌入式系统的代价、最后还需要考虑嵌入式系统是针对于群组问题，不是单独存在。

<三方面分析>

嵌入式系统是以提高对象体系智能性、控制力和人机交互能力为目的，通过相互作用和内在指标评价的，嵌入到对象体系中的专用计算机系统。智能性指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；控制力指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；人机与群体交互指在交互设备、交互软件与人的协调下，进行双向的信息交换以及嵌入式系统是以群体而非单个为节点进行研究、节点在接下来的工作中可以直接对接云计算等。

第二章章节问题

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？

(1) 看问题的角度上

层次结构图是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的特点,体现了一种系统的思想,是由内到外的。对象图仅仅是逻辑意义上的,反映了系统的应用角度与内在形态,在逻辑上是完整的,也是系统的,但相对于五层结构来说,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) 在层次间的关系上

层次结构图是一个层次分明、结构严谨的框架,而且具有自己的约束条件.层次结构图的最底层是硬件层,最高层是应用层,各层的顺序是严格固定的,每一层只与它相邻的层发生关系.对象图是一个相对松散的结构,各模块之间是并列的关系,每两模块之间都具有联系,这也符合其逻辑应用的侧重点。

(3) 侧重点功能方面

层次结构图侧重于嵌入式系统整体的设计与分析,而对象化图则侧重于逻辑应用。换句话说,对象图更方便对嵌入式产品的分析与评估,体现了对嵌入式系统的定义。

2. 控制数学模型与广义数学模型的异同处？

控制数学模型是描述系统变量间相互关系的动态性能的运动方程,其中控制的实质是检测偏差和纠正偏差; 广义数学模型包含控制数学模型,即控制数学模型是广义数学模型的一个分支,依照不同的功能与项目,驱动具体的数学模型,例如控制数学模型。

3. 如何理解嵌入式系统设计的三法则？

(1) 对象追溯·除法法则：通过对本元的追溯以及本元之间的交叉以提高嵌入式系统的简介刚性度；

(2) 系统赋能·乘法法则：在稳定的同时不断增强，即固本增强，此外，增加群组协作，以提高自治互助度；

(3) 对象进化·加法法则：进行自我排异、不断回滚包容，以增强嵌入式系统的体验稳定度。

第三章 章节问题

1. 嵌入式微处理器指令集 5 种类型，说明其中 CISC&RISC，与进程&线程之间的相似关系，并鉴别特斯拉电动车小电池积聚成大动力电池，是否也是这样的逻辑？

<相似关系>

1、CISC&RISC:

CISC 是“复杂指令系统计算机”，RISC 是“精简指令集计算机”。在 CISC 中指令是按照顺序串行执行的，比较简单但是速率较慢。其指令系统比较丰富，相比 RISC，CISC 执行复杂任务的效率较高。Intel 的 x86、Windows 操作系统等都运用的是 CISC 指令集。与 CISC 相比，RISC 的指令类型较少，执行速度更快，对于复杂指令通常使用组合指令实现，所以在复杂任务上效率较低。IBM 公司的 Power PC、HP 公司的 PA-RISC 等都使用的是 RISC 指令集。

2、进程&线程:

一个线程只能属于一个进程，但一个进程可以有很多线程。统一进程上的所有线程可以同时使用该进程的全部资源。多个进程可以并行，统一进程中的不同线程也可以并行。比如可以在同一进程中建立多个线程，其中一个负责某功能的调用，另一线程负责页面的响应刷新。

<鉴别逻辑>

在将特斯拉电动车小电池积聚成大动力电池的过程中，可以将这一过程与嵌入式微处理器指令集和进程/线程之间的相似关系进行类比。

1、CISC 与 RISC:

在这个情境下，可以将小电池比喻为 CISC（复杂指令系统计算机），而动力电池则类似于 RISC（精简指令集计算机）。小电池可能是较为复杂的单体电池，具有多个组件和功能。

然而，将这些小电池积聚成大动力电池时，可能采用更简单、更高效的设计和组装方式，类似于 RISC 指令集的设计思想。

2、进程与线程：

将一个电动车的整体生产过程比作一个进程，而在这个进程中，小电池的组装过程可以看作是多个线程的并行执行。一个电动车的生产过程是一个整体的进程，而在这个过程中，可以并行进行多个任务，就像在一个进程中可以同时运行多个线程一样。其中，每个线程（或者说每个小电池的组装过程）都可以独立执行，并且共享该进程的资源，从而提高了生产效率和响应速度。

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？

可分为系统时钟和 RTC。系统时钟可以通过相对独立的 pll 形成多套系统时钟，如主 CPU 时钟、系统总线时钟、内存总线时钟、GPU 单元时钟等。处理器、显示系统、外设等需要在系统时钟的驱动下完成各种指令，如数据发送、A/D 转换等。统一指令的执行起点，使系统能够以群组形式正常运行。RTC 可以实现分频，将预分频器设为较高的值，可以降低功耗、提高效率来提升系统的群组能力。

3.如何理解 CMOS 工艺数字器件公式 $P \propto CV^2F$ ？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？

<公式 $P \propto CV^2F$ >

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。 $P \propto CV^2F$ 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。因为功耗与电压成指数幂的关系，而与时钟频率只是线性关系。所以降低 CMOS 电路的工作电压比降低时钟频率更能降低活动功耗，应该尽可能选定满足电子系统中所有模块的最低要求电压以达到降低能耗的目的。

<锁相环原理>

能够使得外部信号和内部独立时钟同步的功能模块。是将参考信号与受控振荡器输出信号之间的相位进行比较，产生相位误差电压来调整受控振荡器输出信号的相位，从而使受控振荡器输出频率与参考信号频率相一致。

ARM 专题 3 问

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？

ARM 公司不制造自己的芯片，只设计芯片。将知识产权交给 Intel 等生产芯片的公司，在授权过程中收取授权费，后 Intel、三星等生产芯片的公司制造芯片，每制造出一张芯片，ARM 公司向其收取一部分版税。

芯片授权制造的难点在于制备硬件的软件初期投入较高，需要较大的智力投入；核心技术过硬，融合包容性较强；

2. ARM 架构 Vs 处理器交织滚动发展的价值是什么？

ARM 架构是 ARM 处理器的抽象，ARM 处理器是 ARM 结构的具象；

ARM 处理器随 ARM 架构的变化不断弹性进行配合，相互促进共同推进 ARM 架构、处理器的性能，满足更多需要。

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？

<价值>

主要体现在 ARM 处理器异常时对出现计算处理运行问题的细分应对和 ARM 处理器异常是对传统中断的扩充。

ARM 处理器的异常机制在计算架构中的价值在于提供了一种灵活的处理器状态管理方式，可以有效应对计算和处理运行中的各种问题。这种异常机制实际上是对传统中断概念的扩展和增强，使得处理器能够更加精细地响应不同类型的事件和条件。

传统中断主要用于处理外部事件（如硬件中断），而异常机制不仅可以用于处理外部事件，还可以处理内部事件，例如程序执行中的错误或者异常情况（比如除零错误、内存访问越界等）。ARM 处理器的异常机制允许将不同的异常归类，并为每种异常提供不同的处理方式，从而提高了系统的可靠性、稳定性和安全性。

<ARMv9 在 AI 指令增加的体现>

1、AI 专用指令集扩展：ARMv9 架构可能会引入新的指令集扩展，专门用于加速人工智能相关计算任务。这些指令集可能包括针对神经网络推理、深度学习算法等方面的优化指令，以提高 AI 计算的效率和性能。

2、向量处理单元优化：AI 应用通常涉及大规模数据处理，ARMv9 架构可能对其向量处理单元进行优化，以更好地支持向量化运算，提高并行计算能力。

3、加速器支持：ARMv9 架构可能会增强对特定 AI 加速器的支持，包括更好的协处理

器接口、更高效的加速器访问机制等，以提升 AI 计算的整体性能。

4、新的数据类型和操作指令：ARMv9 架构可能引入新的数据类型和操作指令，以更好地支持 AI 计算中常见的数据格式和运算需求，例如浮点数格式、矩阵运算等。

第四章章节问题

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？

ARM 虚拟地址空间到物理存储空间的映射主要由存储管理单元 MMU 完成，使用位于内存中的一个对照表，即页表来实现此功能。ARM 系统中 CP15 的 C2 寄存器用来保存页表的基地址。地址转换的过程实际是查询页表的过程，为加快查询速度，通常使用快表查询。映射是以内存块为单位进行的。根据存储块大小，可以支持多种地址变换，ARM 支持的存储块的大小为：段、大页、小页、微页。在 MMU 中采用两级页表实现地址映射的过程。

little 多字节数据存储方式采用小端字节序，将低序字节存储在起始地址（低位编址），big 多字节数据存储方式采用大端字节序，将高序字节存储在起始地址（高位编址），也就是和日常使用的顺序一致。

例子：

低地址→高地址

little CDH ABH 34H 12H

big 12H 34H ABH CDH

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？

<BIOS 和 ISA 总线作用>

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 的引导驱动层的作用>

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. 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？

<启动过程>

stage1 第一阶段的功能：硬件设备初始化；加载 U-Boot 第二阶段代码到 RAM 空间；设置好栈；跳转到第二阶段代码入口

stage2 第二阶段的功能：初始化本阶段使用的硬件设备；检测系统内存映射；将内核从 Flash 读取到 RAM 中；为内核设置启动参数；调用内核

<最小系统的分类>

最小系统包括裸机最小系统、OS 内核最小系统、OS 最小系统。裸机状态即指没有任何多余的东西，存在即有自己的行为，能执行基本的功能，类似于人刚出生；OS 内核最小系统即指在裸机的基础上，加上 Linux 等文件系统完成的最小系统，即初步实现一定的认知，类似于初步接受教育；最小系统即指能够完成基本的功能，类似于完成义务教育。最小系统以是否具有操作系统为标准，可以分为：不带操作系统的系统及有操作系统的系统。