计算机系统结构

- 概论
  - 计算机系统的层次结构
    - 现代通用计算机系统的分级
      - 层级结构（由高到低）
        - 应用语言及其级
        - 高级语言机器级
        - 汇编语言机器级
        - 操作系统机器级
        - 大多使用软件实现
        - 传统机器语言机器级
        - 微程序（固件）实现
        - 微程序机器级
        - 硬件实现
    - 现代通用计算机系统的相对位置
    - 各机器级的实现所用的翻译技术和解释技术
      - 翻译技术：先用转换程序将高一级机器级上的程序整个地编号称低一级机器级上等效的程序，然后再低一级机器上实现的技术
      - 解释技术：再低级机器级上用它的一串语句或者指令来仿真高级机器级上的一条语句或者指令的功能，是通过对高级机器语言程序中的每条语句或指令逐条解释来实现的技术
  - 计算机系统结构、组成和实现
    - 计算机系统结构、组成和实现的定义和研究方面
      - 计算机系统结构的定义：从计算机的层级结构角度来看，系统结构是对计算机系统中各级界面的定义及其上下的功能分配，每集都有其自己的系统结构。
      - 计算机系统结构的研究方面
        - 软、硬件直接的功能分配
        - 对传统机器级界面的确定
    - 计算机系统结构是软、硬件的主要界面
    - 系统结构、组成和实现的关系和影响
    - 透明性概念，对具体问题能正确给出是否应透明的选择
  - 计算机系统的软、硬件取舍及定量设计原理
    - 软、硬件实现的优缺点：一般来说，提高硬件功能的比例可提高解题速度，减少程序所需的存储空间，但会增加硬件成本，降低硬件利用率和计算机系统的灵活性及适应性；而提高软件功能的比例可降低硬件成本，提高系统的灵活性、适应性，但解题速度会下降，软件设计费用和所需存储器用量增加。
    - 软、硬件取舍的基本原则
      - 应考虑现有硬、器件（主要是逻辑器件和存储器件）条件下，系统要有高的性能价格比，主要从现实费用、速度和其他性能要求来综合考虑
      - 要考虑准备采用和可能采用的组成技术，使之尽可能不要过多或不合理地限制各种组成、实现技术的应用。
      - 不能仅从“硬”的角度考虑如何便于应用组成技术的成功和便于发挥器件技术的进展，还应从“软”的角度把如何为编译和操作系统的实现以及为高级语言程序的设计提供更多、更好的硬件支持放在首位。
    - 计算机系统的定量设计原理
      - 哈夫曼压缩原理：尽可能加速处理高概率事件远比加速处理概率很低的事件对性能的提高要显著
      - Amdahl定律
      - 程序访问的局部性定律
        - 时间上：时间上的局部性能是指现在正使用的信息可能不久还要使用，这是因为程序存在着循环
        - 空间上：空间上的局部性是指最近的将来要用到的信息很可能与现在正在使用的信息在程序位置上是近邻的，这是因为指令通常是顺序存放、顺序执行的，数据也通常是以向量、陈列、树、表等形式聚集地放在一起。
    - 计算机系统的3种设计思路和存在的问题
    - 计算机系统“由中间开始”设计的方法和优点
  - 软件、应用、器件的发展对系统结构的影响
    - 软件可移植性的定义、实现途径和原因
      - 可移植性的定义：软件的可移植性指的是软件不修改或只修经少量修改就可以由一台及其移到另一台机器上运行，同一软件可应用于不同的环境。
      - 实现软件移植的技术
        - 统一高级语言
        - 采用系列机
        - 模拟和仿真
    - 采用统一高级语言实现软件移植性的方法、使用场合、存在问题和对策
    - 采用系统机实现软件移植性的方法、使用场合、存在问题和对策
    - 软件向前、向后、向下、向上兼顾定义，系列机对软件兼容的要求
    - 系列机发展型号及其的可取做法
    - 采用模拟与仿真实现软件移植性的方法、使用场合、优点、存在问题和对策
      - 模拟与仿真
    - 软件移植手段综述
    - 器件发展对系统结构的影响
      - 非用户片、现场片、用户片的定义
        - 非用户片：非用户片也称为通用片，其功能是由器件厂家产生时固定的，器件的用户只能使用，不能改变器件内部功能
      - 器件发展是推动系统结构和组成前进的因素
      - 器件发展如何改变逻辑设计的方法
  - 系统结构种的并行性开发及计算机系统的分类
    - 并行性的概念与开发
      - 并行性定义、二重含义及开发途径
      - 并行性等级划分
        - 指令内部：一条指令内部各个微操作之间的并行执行
        - 指令之间：多条指令的并行执行
        - 任务或进程之间：多个任务或程序段的并行执行
        - 作业或程序之间：多个作业或多道程序的并行执行
    - 沿3种并行性开发途径的多机系统类型和特点
      - 时间重叠 =
      - 资源重复 =
      - 资源共享 =
      - 耦合度概念
    - 耦合度概念
    - 计算机系统弗林分类法
- 数据流计算机和归约机
- 多处理机
- 向量处理机
- 标量处理机
- 数据表示、寻址方式与指令系统
  - 数据表示
    - 高级数据表示
      - 自定义数据表示
        - 标识符
          - 优点
            - 简化了指令系统和程序设计。由于指令通用于多种数据类型的处理，减少了指令系统中指令的种类，因此简化了程序设计。
            - 简化了编译程序。
            - 便于执行一致性校验
            - 能由硬件自动变换数据类型
            - 支持数据库系统的实现于数据类型无关的要求，是程序不用修改即可处理多种不同类型的数据
            - 为软件调试和应用软件开发提供了支持。
        - 数据描述符
      - 向量、数组数据表示
      - 堆栈数据表示
      - 子主题 4
    - 引入数据表示的原则
    - 子主题 3
  - 寻址方式
  - 指令系统的设计和优化
  - 指令系统的发展和改进
- 存储、中断、总线与I/O系统
- 存储体系