

# 第1小时

## 计算机系统基础知识

### 1.0 章节考点分析

第1小时主要学习计算机硬件基础知识、计算机软件基础知识、计算机语言、多媒体技术等内容。

根据考试大纲，本小时知识点涉及单项选择题，按以往全国计算机技术与软件专业技术资格（水平）考试的出题规律约占2~6分。本小时内容属于基础知识范畴，一般不会在案例分析题中出现。本小时知识架构如图1.1所示。

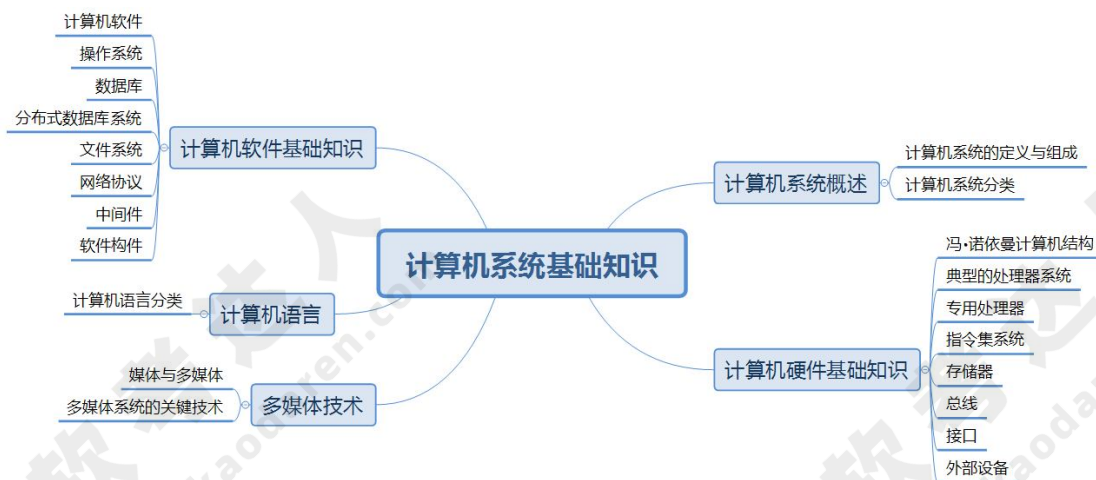


图 1.1 本小时知识架构



## 【导读小贴士】

想成为一名合格的高级架构师，需要在计算机领域中“上知天文，下知地理”，所谓“万丈高楼平地起”，本小时所要讲述的计算机系统基础知识，属于需要掌握的庞杂的知识域中的一小部分，都是入门的基础知识，教材上言简意赅，这里我们会对教材进行补充，加以整理，力求让不同阶段的读者都能有所收益。

## 1.1 计算机系统概述

### 【基础知识点】

#### 1. 计算机系统的定义与组成

计算机系统（Computer System）是指用于数据管理的计算机硬件、软件及网络组成的系统。计算机系统可划分为硬件（子系统）和软件（子系统）两部分。硬件由机械、电子元器件、磁介质和光介质等物理实体构成；软件是一系列按照特定顺序组织的数据和指令，并控制硬件完成指定的功能。

#### 2. 计算机系统分类

计算机系统的分类维度很多，也较为复杂，可以从硬件的结构、性能、规模上划分，亦可从软件的构成、特征上划分，或者从系统的整体用途、服务对象等进行分类。

## 1.2 计算机硬件基础知识

### 【基础知识点】

#### 1. 冯·诺依曼计算机结构

冯·诺依曼计算机结构将计算机硬件划分为运算器、控制器、存储器、输入设备、输出设备 5 个部分。但在现实的硬件中，控制单元和运算单元被集成为一体，封装为通常意义上的中央处理器（Central Processing Unit，CPU）。

#### 2. 典型的处理器系统

典型的处理器系统结构如图 1.2 所示。

#### 3. 专用处理器

除了通用的处理器，用于专用目的的专用处理器芯片不断涌现，常见的有图形处理器（Graphics Processing Unit，GPU）、信号处理器（Digital Signal Processor，DSP）以及现场可编程逻辑门阵列（Field Programmable Gate Array，FPGA）等。GPU 常有数百个或数千个内核，经过优化可并行运行大量计算；DSP 专用于实时的数字信号处理，常采用哈佛体系结构。

4. 指令集系统

典型的处理器根据指令集的复杂程度可分为复杂指令集（Complex Instruction Set Computers, CISC）与精简指令集（Reduced Instruction Set Computers, RISC）两类。CISC 以 Intel、AMD 的 x86 CPU 为代表，RISC 以 ARM 和 Power 为代表。国产处理器目前有龙芯、飞腾、申威等品牌，常采用 RISC-V、MIPS、ARM 等精简指令集架构。

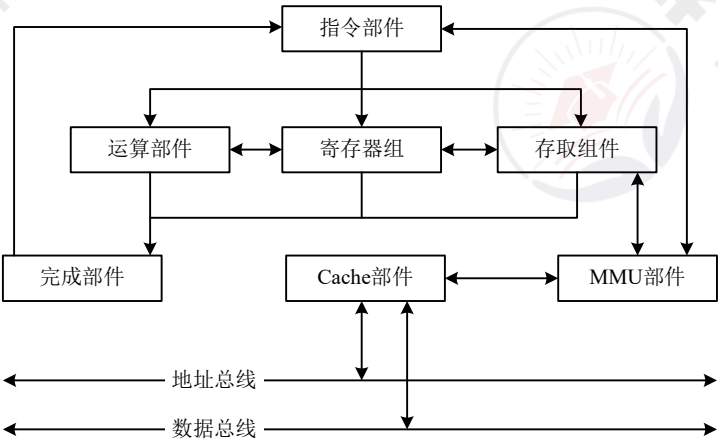


图 1.2 典型的处理器体系结构示意图

5. 存储器

存储器是利用半导体、磁、光等介质制成用于存储数据的电子设备。根据存储器的硬件结构可分为 SRAM、DRAM、NVRAM、Flash、EPROM、Disk 等。按照与处理器的物理距离可分为 4 个层次：片上缓存、片外缓存、主存（内存）、外存。其访问速度依次降低，而容量依次提高。

6. 总线

总线（Bus）是指计算机部件间遵循某一特定协议实现数据交换的形式，即以一种特定格式按照规定的控制逻辑实现部件间的数据传输。按照总线在计算机中所处的位置划分为内总线、系统总线和外部总线。目前，计算机总线存在许多种类，常见的有并行总线和串行总线。两者的区别见表 1.1。

表 1.1 并行总线与串行总线的区别

| 名称   | 数据线             | 特点            | 应用                    |
|------|-----------------|---------------|-----------------------|
| 并行总线 | 多条双向数据线         | 有传输延迟，适合近距离连接 | 系统总线（计算机各部件）          |
| 串行总线 | 一条双向数据线或两条单向数据线 | 速率不高，但适合长距离连接 | 通信总线（计算机之间或计算机与其他系统间） |

7. 接口

接口是指同一计算机不同功能层之间的通信规则。计算机接口有多种，常见的有输入输出接口

第 1 小时

如 HDMI、SATA、RS-232 等；网络接口如 RJ45、FC 等；以及 A/D 转换接口等非标准接口。

#### 8. 外部设备

外部设备也称为外围设备，是计算机结构中的非必要设备，但从功能上又常常不可缺少，例如键盘、鼠标、显示器等。虽然种类多样，但都是通过接口实现与计算机主体的连接，并通过指令、数据实现预期的功能。

### 1.3 计算机软件基础知识

#### 【基础知识点】

##### 1. 计算机软件

计算机软件是指计算机系统上的程序及其文档，是计算任务的处理对象和处理规则的描述。软件系统是指在计算机硬件系统上运行的程序、相关的文档资料和数据集合。计算机软件可用来扩充计算机系统的功能，提高计算机系统的效率。按照软件所起的作用和需要的运行环境不同，通常将计算机软件分为系统软件和应用软件两大类。

(1) 系统软件。为整个计算机系统配置的不依赖特定应用领域的通用软件，对计算机系统的硬件和软件资源进行控制和管理，并提供运行服务支持。

(2) 应用软件。是指为某类应用需要或解决某个特定问题而设计的软件，常与具体领域相关联，如教学软件。

##### 2. 操作系统

操作系统是计算机系统的资源管理者，包含对系统软、硬件资源实施管理的一组程序。操作系统通常由操作系统的内核（Kernel）和其他许多附加的配套软件所组成，如用户界面、管理工具、开发工具和常用应用程序等。操作系统的重要作用如下：

(1) 管理计算机中运行的程序和分配各种软、硬件资源。

(2) 为用户提供友善的人机界面。

(3) 为应用程序的开发和运行提供一个高效率的平台。

操作系统具有并发性、共享性、虚拟性和不确定性的特征。

操作系统的分类：

(1) 批处理操作系统，根据同时执行的作业数又分为单道批处理和多道批处理。一个作业由用户程序、数据和作业说明书（作业控制语言）3 个部分组成。

(2) 分时操作系统，将 CPU 的工作时间划分为许多很短的时间片，每个时间片分别为一个终端的用户提供服务或者执行一个作业。分时系统主要有 4 个特点：多路性、独立性、交互性和及时性。

(3) 实时操作系统，对于外来信息能够以足够快的速度进行处理，并在被控对象允许的时间范围内快速做出反应，对可靠性要求很高，并且不强制要求用户交互。实时系统的应用非常广泛。

(4) 网络操作系统，使联网计算机能有效地共享网络资源，为网络用户提供各种服务和接口。特征包括硬件独立性和多用户支持等。

(5) 分布式操作系统，指为分布式计算机系统配置的操作系统。分布式操作系统是网络操作系统的更高级形式，它保持网络系统所拥有的全部功能，同时又有透明性、可靠性和高性能等特性。

(6) 嵌入式操作系统，运行在嵌入式智能设备环境中，对整个智能硬件以及它所操作、控制的各种部件装置等资源进行统一协调、处理、指挥和控制，特点是微型化、可定制、可靠性和易移植性。常采用硬件抽象层（Hardware Abstraction Layer, HAL）和板级支撑包（Board Support Package, BSP）来提高易移植性，常见的嵌入式实时操作系统有 VxWorks、 $\mu$ Clinux、PalmOS、WindowsCE、 $\mu$ C/OS-II 和 eCos 等。

### 3. 数据库

数据库（DataBase, DB）是指长期存储在计算机内、有组织的、统一管理的相关数据的集合。数据是按一定格式存放的，具有较小的冗余度、较高的数据独立性和易扩展性，可为多个用户共享。数据库可以分为：关系型数据库、键值（Key-Value）数据库、列存储数据库、文档数据库等。这里只做简单介绍，详细内容见第 8 小时。

### 4. 分布式数据库系统

分布式数据库系统（Distributed DataBase System, DDBS）是针对地理上分散，而管理上又需要不同程度集中的需求而提出的一种数据管理信息系统。满足分布性、逻辑相关性、场地透明性和场地自治性的数据库系统被称为完全分布式数据库系统。分布式数据库系统的特点是数据的集中控制性、数据独立性、数据冗余可控性、场地自治性和存取的有效性。

### 5. 文件系统

文件（File）是具有符号名的、在逻辑上具有完整意义的一组相关信息项的集合。文件系统是操作系统中实现文件统一管理的一组软件和相关数据的集合，是专门负责管理和存取文件信息的软件机构。

文件的类型如下：

- (1) 按性质和用途分类可将文件分为系统文件、库文件和用户文件。
- (2) 按信息保存期限分类可将文件分为临时文件、档案文件和永久文件。
- (3) 按保护方式分类可将文件分为只读文件、读/写文件、可执行文件和不保护文件。
- (4) UNIX 系统将文件分为普通文件、目录文件和设备文件（特殊文件）。

文件的存取方法：通常有顺序存取和随机存取两种方法。

文件组织方法：有连续结构、链接结构和索引结构，还有多重索引方式。

文件存储空间的管理知道存储空间的使用情况，空间管理的数据结构通常称为磁盘分配表（Disk Allocation Table），有空闲区表、位示图和空闲块链 3 种。位示图用每一位的 0 和 1 表示一个区块空闲或被占用，如图 1.3 所示。

### 6. 网络协议

常用的网络协议包括局域网协议（Local Area Network, LAN）、广域网协议（Wide Area Network, WAN）、无线网协议和移动网协议。互联网使用的是 TCP/IP 协议簇。本知识点的详细内容将在第 3 小时中讲解。



|         | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---------|---|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| 第 1 个字  | 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0  | 0  | 1  | 1  | 1  | 0  |
| 第 2 个字  | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 1 | 1 | 0  | 0  | 0  | 0  | 1  | 1  |
| 第 3 个字  | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 1 | 0 | 0 | 0  | 0  | 1  | 1  | 1  | 1  |
| 第 4 个字  |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| ⋮       |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |
| 第 15 个字 |   |   |   |   |   |   |   |   |   |   |    |    |    |    |    |    |

图 1.3 位示图例

## 7. 中间件

中间件（Middleware）是应用软件与各种操作系统之间使用的标准化编程接口和协议，是基础中间件（分布式系统服务）软件的一大类，属于可复用软件的范畴。

常见中间件的分类如下：

- （1）通信处理（消息）中间件，保证系统能在不同平台之间通信，例如 MQSeries。
- （2）事务处理（交易）中间件，实现协调处理顺序、监视和调度、负载均衡等功能，例如 Tuxedo。
- （3）数据存取管理中间件，为不同种类数据的读写和解密提供统一的接口。
- （4）Web 服务器中间件，提供 Web 程序执行的运行时容器，例如 Tomcat、JBoss 等。
- （5）安全中间件，用中间件屏蔽操作系统的缺陷，提升安全等级。
- （6）跨平台和架构的中间件，用于开发大型应用软件。
- （7）专用平台中间件，为解决特定应用领域的开发设计问题提供构件库。
- （8）网络中间件，包括网管工具、接入工具等。

## 8. 软件构件

构件又称为组件，是一个自包容、可复用的程序集，这个集合整体向外提供统一的访问接口，构件外部只能通过接口来访问构件，而不能直接操作构件的内部。构件的两个最重要的特性是自包容与可重用，利用软件构件进行搭积木式地开发。优点：易扩展、可重用、并行开发。缺点：需要经验丰富的设计师、快速开发与质量属性之间需要妥协、构件质量影响软件整体的质量。

商用构件的标准规范有：

- （1）OMG 的公共对象请求代理架构（Common Object Request Broker Architecture，CORBA）是一个纯粹的规范而不是产品，主要分为 3 个层次：对象请求代理（Object Request Broker，ORB）、公共对象服务和公共设施。采用 IDL 定义接口，并易于转化为具体语言实现。
- （2）SUN 的 J2EE，定义了完整的基于 Java 语言开发面向企业分布的应用规范，其中 EJB 是 J2EE 的构件标准，EJB 中的构件称为 Bean，可以分为会话 Bean、实体 Bean 和消息驱动 Bean。
- （3）Microsoft 的 DNA 2000，采用 DCOM/COM/COM+作为标准的构件。

## 1.4 计算机语言

### 【基础知识点】

计算机语言（Computer Language）是指人与计算机之间用于交流的一种语言，主要由一套指令组成，而这套指令一般包括表达式、流程控制和集合三大部分内容。

计算机语言的分类有：

（1）机器语言。机器语言是第一代计算机语言，是计算机自身具有的“本地语”，由计算机所能直接理解和执行的所有指令组成。指令格式由操作码和操作数两部分组成。

（2）汇编语言。汇编语言在机器语言的基础上采用英文字母和符号串来表达指令，是机器语言的符号化描述。每条语句均由名字、操作符、操作数和注释 4 个字段（Fields）组成。伪指令语句包括数据定义伪指令 DB、DW、DD，段定义伪指令 SEGMENT，过程定义伪指令 PROC 等，编译后不产生机器代码。

（3）高级语言。高级语言比汇编语言更贴近于人类使用的语言，易于理解、记忆和使用。常见的高级语言包括 C、C++、Java、Python 等。

（4）建模语言。建模语言主要指的是统一建模语言（Unified Modeling Language, UML），UML 由 3 个要素构成：UML 的基本构造块（事物、关系）、图（支配基本构造块如何放置在一起的规则）和运用于整个语言的公用机制。

1) 事物。UML 中有 4 种事物：结构事物、行为事物、分组事物和注释事物。

a. 结构事物：名词、静态部分，用于描述概念或物理元素。结构事物包括类（Class）、接口（Interface）、协作（Collaboration）、用例（UseCase）、主动类（Active Class）、构件（Component）、制品（Artifact）和节点（Node），如图 1.4 所示。

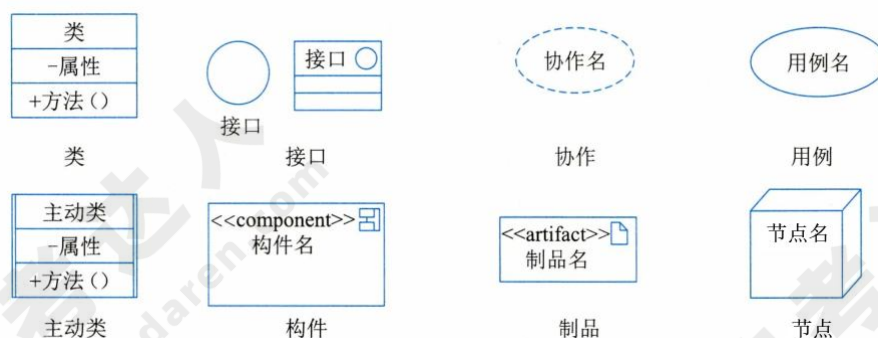


图 1.4 结构事物

b. 行为事物：动词，描述了跨越时间和空间的行为。行为事物包括交互（Interaction）、状态机（State Machine）和活动（Activity），如图 1.5 所示。



图 1.5 行为事物

c. 分组事物：包是最常用的分组事物，结构事物、行为事物甚至其他分组事物都可以放进包内，如图 1.6 所示。

d. 注释事物：注释即注解，用来描述、说明和标注模型的任何元素，如图 1.7 所示。

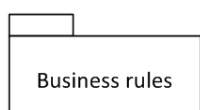


图 1.6 包

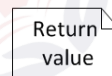
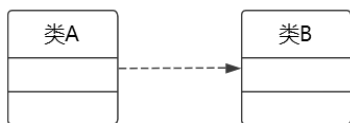


图 1.7 注释

2) 关系。UML 中有 4 种关系：依赖、关联、泛化和实现。4 种关系如图 1.8 所示。

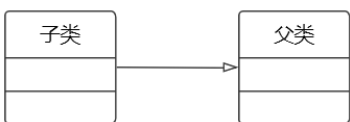
a. 依赖关系。其中一个事物（独立事物）发生变化会影响另一个事物。依赖关系是一种使用的关系。



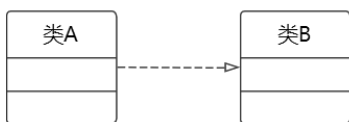
(a) 依赖关系



(b) 关联关系



(c) 泛化关系



(d) 实现关系

图 1.8 UML 中的 4 种关系

b. 关联关系。是一种拥有的关系，关联提供了不同类的对象之间的结构关系，它在一段时间内将多个类的实例连接在一起。一般认为关联关系有 2 个特例：一个是聚合关系，另一个是组合关系。聚合关系表示类之间的整体与部分的关系，其含义是“部分”可能同时属于多个“整体”，“部分”与“整体”的生命周期可以不相同。组合关系也是表示类之间的整体与部分的关系。与聚合关系的区别在于，组合关系中的“部分”只能属于一个“整体”，“部分”与“整体”的生命周期相同，“部分”随着“整体”的创建而创建，也随着“整体”的消亡而消亡。

c. 泛化关系。泛化是一种特殊/一般关系，特殊元素（子元素）的对象可替代一般元素（父元素）的对象。



d. 实现关系。在两种情况下会使用实现关系：一种是在接口和实现它们的类或构件之间；另一种是在用例和实现它们的协作之间。

3) 图。图是一组元素的图形表示，大多数情况下把图画成顶点（代表事物）和弧（代表关系）的连通图。

UML 2.0 提供了 14 种图，分别是类图、对象图、用例图、序列图、通信图、状态图、活动图、构件图、部署图、制品图、组合结构图、包图、交互概览图和计时图（定时图）。序列图、通信图、交互概览图和计时图均被称为交互图。

类图如图 1.9 所示。类图展现了一组对象、接口、协作和它们之间的关系。

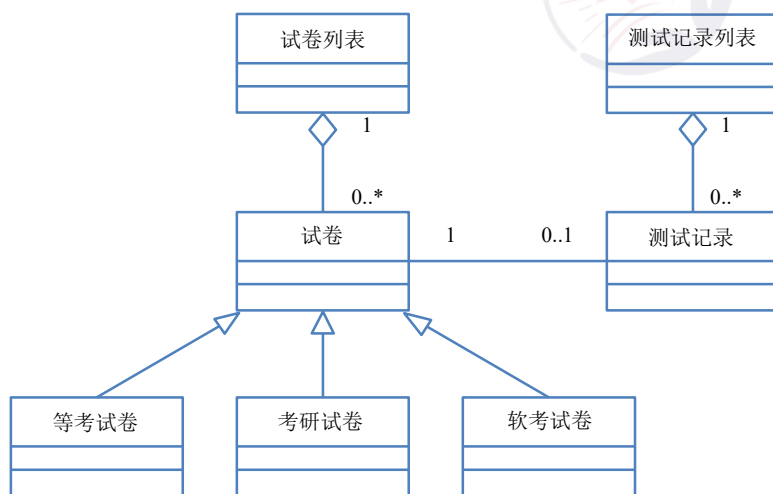


图 1.9 UML 图中的类图

用例图如图 1.10 所示。用例图（Use Case Diagram）展现了一组用例、参与者（Actor）以及它们之间的关系。用例之间有扩展关系（<<extend>>）和包含关系（<<include>>），参与者和用例之间有关联关系，用例与用例、参与者与参与者之间有泛化关系。包含关系的特点是当两个或多个用例中共用一组相同的动作时，可以将这组相同的动作抽出来作为一个独立的子用例，供多个基用例所共享；扩展关系则是对基用例的扩展，基用例是一个完整的用例，即使没有子用例的参与，也可以完成一个完整的功能。

UML 中有 5 种视图（View）：用例视图、逻辑视图、进程视图、实现视图、部署视图，其中的用例视图居于中心地位。

（5）形式化方法和形式化语言。形式化方法是把概念、判断、推理转化成特定的形式符号后，对形式符号表达系统进行研究的方法。形式化方法有不同的分类方法。根据描述方式分，有模型描述和性质描述两类；根据表达能力分，有模型方法、代数方法、进程代数方法、逻辑方法和网络模型方法 5 类。形式化方法的开发过程贯穿软件工程的整个生命周期。

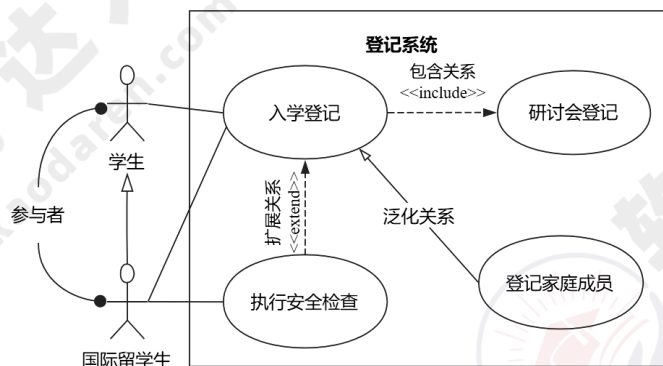


图 1.10 UML 用例图

Z 语言是一种形式化语言，具有“状态—操作”风格，借助模式来表达系统结构。建立于集合论和数理逻辑的基础上，是一个强类型系统，可以使用自然语言。

## 1.5 多媒体技术

### 1. 媒体与多媒体

媒体是承载信息的载体，即信息的表现形式（或者传播形式），如文字、声音、图像、动画和视频等。多媒体有 4 个重要的特征：

- (1) 多维化，即媒体的多样化。
- (2) 集成性，多媒体与设备集成，也与信息和表现集成。
- (3) 交互性，可向用户提供更有效的控制和使用信息的手段。
- (4) 实时性，音频和视频等信息具有很强的时间特性。

多媒体系统通常由硬件和软件组成，其中多媒体硬件主要包括计算机主要配置和外部设备以及和各种外部设备的控制接口；多媒体软件主要包括多媒体驱动软件、多媒体操作系统、多媒体数据处理软件、多媒体创作工具软件和多媒体应用软件等。

### 2. 多媒体系统的关键技术

(1) 视、音频技术，视频技术包括视频数字化和视频编码技术两个方面；音频技术包括音频数字化、语音处理、语音合成及语音识别 4 个方面。

(2) 通信技术，是多媒体系统中的一项关键技术，通常包括数据传输信道技术和数据传输技术。

(3) 数据压缩技术，包括即时压缩和非即时压缩、数据压缩和文件压缩、无损压缩与有损压缩等。

(4) 虚拟现实（Virtual Reality, VR）/增强现实（Augmented Reality, AR）技术，虚拟现实又称人工现实、临境等，是一种可以创建和体验虚拟世界的计算机仿真系统，采用计算机技术生成一个逼真的视觉、听觉、触觉、味觉及嗅觉的感知系统与用户交互；增强现实技术是指把原本在现实世界的一定时间和空间范围内很难体验到的实体信息（视觉信息、声音、味道和触觉等），通过模拟仿真后，再叠加到现实世界中被人类感官所感知，从而达到超越现实的感官体验。VR/AR 技

术主要分为桌面式、分布式、沉浸式和增强式 4 种。

## 1.6 练习题

1. 目前处理器市场中存在 CPU 和 DSP 两种类型的处理器，分别用于不同的场景，这两种处理器具有不同的体系结构，DSP 采用（ ）。

- A. 冯·诺依曼结构                      B. 哈佛结构  
C. FPGA 结构                         D. 与 GPU 相同的结构

解析：编程 DSP 芯片是一种具有特殊结构的微处理器，为了达到快速进行数字信号处理的目的，DSP 芯片一般都采用特殊的软硬件结构：哈佛结构。

哈佛结构将存储器空间划分成两个，分别存储程序和数据。它们有两组总线连接到处理器核，允许同时对它们进行访问，每个存储器独立编址，独立访问。这种安排将处理器的数据吞吐率加倍，更重要的是同时为处理器核提供数据与指令。在这种布局下，DSP 得以实现单周期的 MAC 指令。

在哈佛结构中，由于程序和数据存储器在两个分开的空间中，因此取指和执行能完全重叠运行。

答案：B

2. （ ）是专用于实时的数字信号处理的处理器。

- A. DSP                      B. CUP                      C. GPU                      D. FPGA

解析：DSP 专用于实时的数字信号处理，常采用哈佛体系结构。

答案：A

3. 在线学习系统中，课程学习和课程考试都需要先检查学员的权限，“课程学习”与“检查权限”两个用例之间属于（1）课程学习过程中，如果所缴纳学费不够，就需要补缴学费，“课程学习”与“缴纳学费”两个用例之间属于（2）；课程学习前需要课程注册，可以采用电话注册或网络注册，“课程注册”与“网络注册”两个用例之间属于（3）。

- (1) A. 包含关系      B. 扩展关系      C. 泛化关系      D. 关联关系  
(2) A. 包含关系      B. 扩展关系      C. 泛化关系      D. 关联关系  
(3) A. 包含关系      B. 扩展关系      C. 泛化关系      D. 关联关系

解析：用例之间的关系主要有包含、扩展和泛化 3 类。

1) 包含关系：当可以从两个或两个以上的用例中提取公共行为时，应该使用包含关系来表示它们。“课程学习”与“检查权限”是包含关系。

2) 扩展关系：如果一个用例明显地混合了两种或两种以上的不同场景，即根据情况可能发生多种分支，则可以将这个用例分为一个基本用例和一个或多个扩展用例，这样使描述可能更加清晰。“课程学习”与“缴纳学费”是扩展关系。

3) 泛化关系：当多个用例共同拥有一种类似的结构和行为的时候，可以将它们的共性抽象成为父用例，其他的用例作为泛化关系中的子用例。“课程注册”与“网络注册”是泛化关系。

答案：A B C