

計算機概論教材

目錄

1. 計算機基本概念
2. 計算機硬件組成
3. 計算機軟件
4. 操作系統
5. 計算機網絡
6. 數據庫基礎
7. 編程基礎

1. 計算機基本概念

1.1 計算機的定義

- 計算機是一種能夠自動執行編程指令的電子設備，用於處理數據和信息。其主要功能包括數據輸入、數據處理、數據存儲和數據輸出。

1.2 計算機的歷史

- 第一代：真空管計算機（1940-1956）
 - 主要特點：使用真空管作為主要電子元件，體積大，功耗高，運算速度慢。
 - 代表機型：ENIAC、UNIVAC I。
 - 應用：主要用於軍事和科學計算。
- 第二代：晶體管計算機（1956-1963）

- 主要特點：使用晶體管替代真空管，體積減小，功耗降低，運算速度提高。
- 代表機型：IBM 7090。
- 應用：商業數據處理、科學計算。
- 第三代：集成電路計算機（1964-1971）
 - 主要特點：使用集成電路（IC），計算機性能大幅提升，體積進一步縮小。
 - 代表機型：IBM System/360。
 - 應用：通用計算機，適用於各種應用。
- 第四代：微處理器計算機（1971-至今）
 - 主要特點：使用微處理器，大規模集成電路（VLSI），個人計算機普及。
 - 代表機型：IBM PC、Apple II。
 - 應用：個人電腦、嵌入式系統、服務器等。

1.3 計算機的分類

- 按用途：
 - 通用計算機：適用於多種應用領域，如 PC、工作站、服務器。
 - 專用計算機：針對特定應用設計，如工業控制計算機、嵌入式系統。
- 按規模：
 - 大型計算機：如 IBM Z 系列，用於大型企業、科研機構。
 - 中型計算機：如 IBM AS/400，用於中型企業。
 - 小型計算機：如迷你計算機，介於大型計算機和個人計算機之間。
 - 微型計算機：即個人計算機（PC），如台式機、筆記本電腦。
- 按架構：
 - 單處理器系統：單一 CPU 處理所有任務。

- 多處理器系統：多個 CPU 協同工作，提高計算效率。
- 分佈式系統：多個計算機系統通過網絡協同工作。

2. 計算機硬件組成

2.1 中央處理器（CPU）

- 功能：
 - 執行指令：CPU 從內存中取指令，解碼並執行。
 - 算術運算：進行基本的數學運算，如加、減、乘、除。
 - 邏輯運算：進行邏輯比較，如與（AND）、或（OR）、非（NOT）。
 - 控制運算：控制其他硬件的運行。
- 組成：
 - 控制單元（CU）：負責指令的解碼和控制信號的生成。
 - 算術邏輯單元（ALU）：進行算術和邏輯運算。
 - 寄存器：CPU 內部的高速存儲單元，用於臨時存儲數據和指令。
 - 緩存（Cache）：高速緩存，用於提高數據訪問速度。
- 進階：
 - 多核處理器：將多個處理核心集成到一個 CPU 中，提高多任務處理能力。
 - 超級標量架構：同時執行多條指令，提高指令執行效率。
 - 流水線技術：將指令分解為多個階段，同時處理多條指令的不同階段，提高吞吐量。

2.2 存儲器

- 主存儲器：
 - 隨機存取存儲器（RAM）：揮發性存儲器，用於存儲正在使用的數據和程序。
 - 動態隨機存取存儲器（DRAM）：常見於主內存，容量大，速度較慢。
 - 靜態隨機存取存儲器（SRAM）：常用於緩存，速度快，容量小。
 - 只讀存儲器（ROM）：非揮發性存儲器，存儲固定的程序和數據。
 - 可編程只讀存儲器（PROM）
 - 可擦寫可編程只讀存儲器（EPROM）
 - 電子可擦寫可編程只讀存儲器（EEPROM）
- 輔助存儲器：
 - 硬盤：大容量、非揮發性存儲設備，用於長期存儲數據。
 - 固態硬盤（SSD）：使用閃存技術，速度快，功耗低。
 - 光盤：如 CD、DVD，用於分發和備份數據。
 - 磁帶：用於大規模數據備份和存檔。
- 進階：
 - 虛擬內存：利用輔助存儲器擴展主存儲器容量，實現多任務處理。
 - 記憶體層次結構：將不同速度和容量的存儲設備結合使用，提高整體性能。

2.3 輸入設備

- 鍵盤：用於輸入文本和控制命令。
- 鼠標：用於指示和選擇屏幕上的對象。
- 掃描儀：將紙質文件轉換為數字圖像。
- 麥克風：用於錄製音頻輸入。
- 進階：

- 觸控屏：結合顯示和輸入功能，用於智能手機和平板電腦。
- 虛擬現實設備：如 VR 頭盔和手柄，用於沉浸式交互。

2.4 輸出設備

- 顯示器：顯示計算機處理結果的屏幕。
 - 液晶顯示器（LCD）
 - 發光二極管顯示器（LED）
 - 有機發光二極管顯示器（OLED）
- 打印機：將數字文檔轉換為紙質文檔。
 - 激光打印機
 - 噴墨打印機
- 音箱：播放音頻輸出。
- 進階：
 - 3D 打印機：將數字模型轉換為實體物品，用於原型製作和生產。
 - 增強現實設備：如 AR 眼鏡，將虛擬信息疊加在現實世界上。

2.5 主板和總線

- 主板：
 - 承載 CPU、存儲器及其他硬件的主要電路板。
 - 提供各種接口和插槽，用於連接各種外部設備。
- 總線：
 - 數據總線：傳輸數據。
 - 地址總線：傳輸地址信息。
 - 控制總線：傳輸控制信號。

- 進階：

- PCI Express 總線：高速數據傳輸總線，用於連接顯卡、存儲設備等。

- USB 總線：通用串行總線，用於連接外部設備，如鍵盤、鼠標、存儲設備。

3. 計算機軟件

3.1 系統軟件

- 操作系統：

- 功能：管理硬件資源、提供用戶界面、文件管理、進程管理、記憶體管理。

- 示例：Windows、Linux、MacOS。

- 驅動程序：

- 功能：使操作系統能夠

與硬件設備通信。

- 示例：顯卡驅動、打印機驅動。

- 進階：

- 虛擬化技術：如 VMware、VirtualBox，允許在一台物理計算機上運行多個虛擬機。

- 容器技術：如 Docker，提供輕量級虛擬化，實現應用的隔離和部署。

3.2 應用軟件

- 辦公軟件：

- 功能：文檔編輯、數據處理、演示文稿製作。
- 示例：Microsoft Office、Google Docs。
- 圖像處理軟件：
 - 功能：圖像編輯和處理。
 - 示例：Adobe Photoshop、GIMP。
- 瀏覽器：
 - 功能：訪問和瀏覽互聯網。
 - 示例：Chrome、Firefox、Safari。
- 進階：
 - 企業資源計劃（ERP）系統：如 SAP、Oracle ERP，用於企業管理和資源規劃。
 - 客戶關係管理（CRM）系統：如 Salesforce，用於管理客戶關係和銷售流程。

4. 操作系統

4.1 操作系統的功能

- 管理硬件資源：
 - CPU 管理：進程調度、多任務處理。
 - 記憶體管理：內存分配和回收、虛擬內存。
- 提供用戶界面：
 - 圖形用戶界面（GUI）：窗口、圖標、菜單。
 - 命令行界面（CLI）：命令提示符、Shell。
- 文件管理：

- 文件系統：文件的創建、刪除、讀取、寫入。
- 文件夾：文件的組織和管理。
- 進程管理：
 - 進程的創建和終止。
 - 進程間通信和同步。
- 記憶體管理：
 - 內存分配和回收。
 - 虛擬內存：內存擴展技術。
- 進階：
 - 安全管理：用戶認證、權限管理、數據加密。
 - 網絡功能：網絡協議支持、網絡資源管理。

4.2 常見操作系統介紹

- Windows：
 - 特點：圖形用戶界面、廣泛應用於個人電腦、支持大量應用程序。
 - 版本：Windows 10、Windows 11。
 - 進階：Windows Server，用於企業級應用和服務器管理。
- Linux：
 - 特點：開源、靈活、廣泛應用於服務器、支持多種發行版（如 Ubuntu、Fedora）。
 - 核心：Linux 內核。
 - 進階：Docker、Kubernetes，用於容器化應用和集群管理。
- MacOS：
 - 特點：Apple 公司開發、專為 Mac 電腦設計、圖形界面優雅。

- 版本：macOS Monterey、macOS Big Sur。
- 進階：iOS 和 iPadOS，基於 macOS 內核的移動操作系統。

4.3 操作系統內核

- 單內核：
 - 特點：所有操作系統服務都運行在內核模式下。
 - 優點：性能高。
 - 缺點：穩定性和安全性相對較差。
- 微內核：
 - 特點：只包含最基本的服務，其他服務運行在用戶模式下。
 - 優點：穩定性和安全性高。
 - 缺點：性能相對較低。
- 進階：
 - 混合內核：結合單內核和微內核的特點，提供高性能和高穩定性。
 - exokernel：提供最小化的內核，允許應用程序直接管理硬件資源。

5. 計算機網絡

5.1 計算機網絡的概念

- 連接多台計算機並進行通信的系統。
- 功能：數據共享、資源共享、通信。

5.2 網絡拓撲結構

- 星型：
 - 特點：所有設備通過集線器或交換機連接。
 - 優點：易於擴展和故障診斷。
 - 缺點：集線器或交換機故障會影響整個網絡。
- 總線型：
 - 特點：所有設備共享一條總線。
 - 優點：結構簡單、成本低。
 - 缺點：總線故障會影響整個網絡、擴展性差。
- 環型：
 - 特點：所有設備形成一個環，每個設備都有兩個鄰居。
 - 優點：易於管理、數據傳輸速度快。
 - 缺點：環中任一設備故障會影響整個網絡。
- 網狀：
 - 特點：每個設備都與多個其他設備直接相連。
 - 優點：高冗餘度和可靠性。
 - 缺點：結構複雜、成本高。
- 進階：
 - 混合型拓撲：結合多種拓撲結構的特點，適應不同的應用場景。
 - 區域網絡（LAN）：如 Wi-Fi，用於小範圍內的無線通信。

5.3 網絡類型

- 局域網（LAN）：
 - 特點：覆蓋範圍小（如辦公室、校園）、數據傳輸速度快。
 - 技術：以太網、Wi-Fi。

- 廣域網 (WAN) :
 - 特點：覆蓋範圍大 (如城市、國家)、數據傳輸速度相對較慢。
 - 技術：ATM、Frame Relay、MPLS。
- 都市網 (MAN) :
 - 特點：覆蓋範圍介於 LAN 和 WAN 之間，通常覆蓋一個城市或大都市區。
 - 技術：光纖網絡、無線網絡。
- 進階：
 - 個人區域網 (PAN)：如藍牙，用於個人設備間的短距離通信。
 - 全球網 (GAN)：如衛星通信，用於全球範圍內的數據傳輸。

5.4 網絡協議

- TCP/IP 協議：
 - 特點：互聯網的基礎協議，支持數據包交換和路由。
 - 層次：應用層、傳輸層、網絡層、鏈路層。
- HTTP/HTTPS :
 - 特點：超文本傳輸協議，用於萬維網數據傳輸，HTTPS 增加了安全層 (SSL/TLS)。
- FTP :
 - 特點：文件傳輸協議，用於在網絡上傳輸文件。
- SMTP/POP3 :
 - 特點：電子郵件協議，SMTP 用於發送郵件，POP3 用於接收郵件。
- 進階：
 - DNS：域名系統，將域名轉換為 IP 地址。
 - DHCP：動態主機配置協議，自動分配 IP 地址和網絡配置。

5.5 網絡安全

- 防火牆：
 - 功能：監控和控制進出網絡的流量。
 - 類型：硬件防火牆、軟件防火牆。
- 加密技術：
 - 功能：保護數據在傳輸過程中的機密性。
 - 類型：對稱加密、非對稱加密。
- 數字簽名：
 - 功能：驗證數據的完整性和發送者的身份。
 - 技術：公鑰加密、散列函數。
- 反病毒軟件：
 - 功能：檢測和刪除惡意軟件，保護計算機安全。
 - 示例：Norton、McAfee、Kaspersky。
- 進階：
 - 入侵檢測系統（IDS）：監控網絡活動，檢測潛在的安全威脅。
 - 雲安全：保護雲端數據和應用，確保雲服務的安全性。

6. 數據庫基礎

6.1 數據庫的概念

- 數據庫是一個組織化的數據集合，用於高效存儲和管理數據。
- 功能：數據存儲、

數據檢索、數據更新。

6.2 數據庫管理系統（DBMS）

- 功能：提供數據庫的創建、管理和使用的工具和接口。
- 示例：
 - MySQL：開源關係型數據庫，常用於 Web 應用。
 - PostgreSQL：高級開源關係型數據庫，支持複雜查詢和數據完整性。
 - SQLite：輕量級嵌入式數據庫，適用於移動應用和嵌入式系統。
 - Oracle：商用關係型數據庫，支持大規模企業應用。
- 進階：
 - NoSQL 數據庫：如 MongoDB、Cassandra，用於非結構化數據和大數據處理。
 - 分佈式數據庫：如 Amazon DynamoDB，用於高可用性和水平擴展。

6.3 數據庫模型

- 層次模型：
 - 特點：數據以層次結構組織，父子關係明確。
 - 優點：數據結構清晰，適合描述層次關係。
 - 缺點：靈活性差，難以處理複雜關係。
- 網絡模型：
 - 特點：數據以網絡結構組織，任意節點可以有多個父節點和子節點。
 - 優點：靈活性高，適合處理複雜關係。
 - 缺點：結構複雜，管理困難。

- 關係模型：
 - 特點：數據以表格形式組織，表之間通過鍵建立關係。
 - 優點：靈活性高，易於理解和操作。
 - 缺點：對於某些應用性能較低。
- 進階：
 - 面向對象數據模型：將對象導向概念引入數據庫設計。
 - 文檔數據模型：如 MongoDB，數據以文檔形式存儲，靈活性高。

6.4 SQL 語言

- 基本語法：
 - 特點：結構化查詢語言，用於操作關係型數據庫。
 - 操作：查詢、插入、更新、刪除數據。
- 查詢語句（SELECT）：
 - 功能：從數據庫中檢索數據。
 - 示例：`SELECT * FROM table_name WHERE condition;`
- 插入語句（INSERT）：
 - 功能：向數據庫中插入新數據。
 - 示例：`INSERT INTO table_name (column1, column2) VALUES (value1, value2);`
- 更新語句（UPDATE）：
 - 功能：更新數據庫中的現有數據。
 - 示例：`UPDATE table_name SET column1 = value1 WHERE condition;`
- 刪除語句（DELETE）：
 - 功能：從數據庫中刪除數據。

- 示例：DELETE FROM table_name WHERE condition;
- 進階：
 - 事務管理：確保數據庫操作的原子性、一致性、隔離性和持久性（ACID）。
 - 儲存過程和觸發器：在數據庫內部實現邏輯處理和自動化操作。

7. 編程基礎

7.1 程序設計語言

- 低級語言：
 - 機器語言：直接用二進制碼編寫，運行速度快，但難以理解。
 - 彙編語言：使用符號代表機器碼指令，易於理解，但仍需特定的硬件知識。
- 高級語言：
 - C：高效、靈活，適用於系統編程。
 - Java：面向對象、平台無關性，廣泛應用於企業級應用。
 - Python：簡單易學、強大的標準庫，適用於快速開發和數據分析。
 - JavaScript：主要用於 Web 開發，支持動態交互效果。
- 進階：
 - 函數式編程：如 Haskell、Scala，強調不可變性和純函數。
 - 併發編程：如 Go、Erlang，用於高並發和分佈式系統。

7.2 編程基本概念

- 變量與數據類型：

- 變量：存儲數據的容器。
- 數據類型：整數、浮點數、字符串、布爾值等。
- 運算符與表達式：
 - 運算符：加、減、乘、除、模、比較、邏輯。
 - 表達式：由變量、常量和運算符組成的有效組合。
- 控制結構：
 - 條件語句（**if**）：根據條件執行不同的代碼。
 - 循環語句（**for**、**while**）：重複執行代碼塊。
- 函數與模塊：
 - 函數：執行特定任務的代碼塊，可重用。
 - 模塊：將相關函數和變量組織在一起，便於管理和使用。
- 進階：
 - 錯誤處理：如 **try-catch** 語句，用於捕獲和處理運行時錯誤。
 - 泛型編程：如 **C++** 模板、**Java** 泛型，提高代碼的重用性和靈活性。

7.3 面向對象編程

- 類與對象：
 - 類：定義對象的模版，包括屬性和方法。
 - 對象：類的實例，具有具體的屬性值和方法行為。
- 繼承：
 - 特點：子類繼承父類的屬性和方法，實現代碼重用。
 - 優點：提高代碼的可維護性和可擴展性。
- 多態：
 - 特點：不同類型的對象可以通過相同的接口進行操作。

- 優點：提高代碼的靈活性和可擴展性。
- 封裝：
 - 特點：將數據和操作數據的方法封裝在一起，隱藏內部實現細節。
 - 優點：提高代碼的安全性和可維護性。
- 進階：
 - 抽象類和接口：提供高級抽象，提高代碼的靈活性和可擴展性。
 - 設計模式：如單例模式、工廠模式，解決常見的設計問題，提升代碼質量。