

进程与线程

定义

进程（Process）和线程（Thread）是操作系统中的基本概念，它们是系统进行资源分配和调度的基本单位。

• 进程

进程是系统进行资源分配和调度的独立单位，是应用程序运行的实体。每个进程都有其自己的虚拟地址空间、代码、数据和系统资源。进程之间相互独立，一个进程无法直接访问另一个进程的数据和代码。如果需要进行沟通，必须使用进程间通信（IPC）机制，如管道、信号、套接字、共享内存等。

操作系统使用进程表（或进程控制块PCB）来管理所有进程的运行。这使得操作系统能够进行多任务处理，即允许多个进程看似同时运行（在单核CPU上是通过时间片轮转来实现，而在多核CPU上可以实现真正的并行运行）。

• 线程

线程是进程内的一个实体，是被系统独立调度和分派的基本单位，有时也被称为轻量级进程（Lightweight Process）。一个进程可以包含一个或多个线程。线程与同属一个进程的其他线程共享进程资源，如内存和文件句柄等，但每个线程有自己的寄存器和栈。

线程之间的通信和数据共享更加容易，因为它们共享相同的进程内存空间。不过，这也意味着必须确保线程安全，以防止多个线程同时写入同一资源造成冲突。

• 进程与线程的关系

- 多线程**：在一个进程中可以并发执行多个线程，这些线程可以利用多核处理器的多核能力真正并行执行。
- 资源共享**：相比于进程间通信，线程间的通信和资源共享更为高效，因为它们无需跨进程的上下文切换。
- 依赖性**：线程依赖于进程而存在，一个进程结束时，它的线程也会被迫结束。

• 进程与线程的选择

- 使用多进程时，每个进程都有独立的内存空间，进程之间不会互相影响，但进程创建、销毁和切换的开销较大。
- 使用多线程时，线程之间的创建、销毁和切换的开销较小，但线程之间的冲突需要通过同步机制来控制。

在现代操作系统中，多线程编程是常见的并发编程方法，可以有效利用多核处理器的计算能力，提高程序的执行效率。

进程间通信方式

计算机系统中，进程间通信（Inter-Process Communication, IPC）是允许进程之间传输数据和信号的一组编程接口和技术。进程间通信的几种常见方式如下：

1. **管道 (Pipes) :**
 - **匿名管道 (Anonymous Pipes) :** 通常用于父子进程间的通信，是单向通信。
 - **命名管道 (Named Pipes) :** 可以在不相关的进程间进行双向通信。
2. **信号 (Signals) :**
 - 一个进程可以发送信号给另一个进程来通知它某个事件已经发生。
3. **消息队列 (Message Queues) :**
 - 消息队列允许进程以消息的形式交换数据，数据以队列的形式存储，直到被接收。
4. **共享内存 (Shared Memory) :**
 - 多个进程可以访问同一块内存区域，是最快的IPC形式，但需要处理同步问题。
5. **信号量 (Semaphores) :**
 - 主要用于同步，比如限制对资源的访问，不过也可以通过信号量传递信息。
6. **套接字 (Sockets) :**
 - 用于在网络中的不同机器间通信，但也可以用于同一台机器上的进程间通信。
7. **文件锁定 (File Locking) :**
 - 进程用它来控制对文件的访问。
8. **内存映射 (Memory-mapped Files) :**
 - 将文件或设备映射至内存，可以由多个进程访问。
9. **消息传递接口 (Message Passing Interfaces, MPI) :**
 - 主要用于高性能计算领域，允许多个进程协作执行计算任务。
10. **条件变量 (Condition Variables) :**
 - 通常与互斥锁结合使用，用于多个进程间的同步。
11. **远程过程调用 (Remote Procedure Calls, RPC) :**
 - 允许一个进程调用另一个进程的函数或方法，就像调用本地函数一样。
12. **本地过程调用 (Local Procedure Calls, LPC) :**
 - 类似于RPC，但专门用于同一台机器上的进程间通信。

计算机组成

计算机组成主要指的是组成计算机硬件系统的各个组成部分以及它们之间的关系。一个典型的计算机系统可以分为五个主要部分：中央处理单元（CPU）、内存、输入设备、输出设备和存储设备。以下是对这些部分的详细介绍：

1. 中央处理单元 (Central Processing Unit, CPU)

CPU是计算机的大脑，负责解释和执行程序指令以及控制其他硬件组件。它由以下子部分组成：

- **算术逻辑单元 (Arithmetic Logic Unit, ALU) :** 执行算术运算（如加法、减法）和逻辑运算（如AND、OR）。
- **控制单元 (Control Unit, CU) :** 解析程序指令，协调和控制数据流向其他部件。
- **寄存器:** 提供极小但极快的存储，用于临时保存数据和指令。

2. 内存

内存是CPU进行数据处理时使用的临时存储区域。它通常指随机存取存储器（Random Access Memory, RAM），RAM是一种易失性存储，当电源关闭时其中的数据会丢失。

3. 输入设备

输入设备允许用户与计算机交互，例如键盘、鼠标、触摸屏、扫描器、麦克风等。

4. 输出设备

输出设备用于将信息传递给用户，如显示器、打印机、扬声器等。

5. 存储设备

存储设备用于数据的长期保存。它包括硬盘驱动器（HDD）、固态驱动器（SSD）、光盘（如CD、DVD）和USB存储设备等。

冯诺曼体系结构

五大部件

冯·诺依曼体系结构认为，计算机硬件系统应由以下五大部件组成：

- **存储器（Memory）**：存放程序和数据。

存储器分为 主存储器(内存储器，CPU能直接访问)和 辅助存储器(外存储器，协助主存储器记忆更多的信息，辅助存储器的信息需要导入到主存储器中，才可以被CPU访问)。

主存储器的工作方式是按存储单元的地址进行存取，这种存取方式称为按地址存取方式(相联存储器既可以按照地址寻址，又可以按照内容寻址，为了与传统存储器区别，又称为内容寻址的存储器！)

主存储器是由地址寄存器(MAR)，数据寄存器(MDR)，存储体，时序控制逻辑，地址寄存器存放访问地址，经过地址译码后找到所选的存储单元。数据寄存器，是存储器与其他部件的中介，用于暂存要从存储器读或写的信息。时序控制逻辑用于产生存储器操作所需的各种时序信号。在现代CPU，MAR和MDR是在CPU中的。

- **运算器（Arithmetic Logic Unit, ALU）**：进行数据处理，即算术运算和逻辑运算。运算器的核心单元是算术逻辑单元(ALU)。
- **控制器（Control Unit, CU）**：**解释指令并控制程序执行**。控制器是计算机的指挥中心，有其指挥各部件自动协调地进行工作，现代计算机将运算器和控制器集成到一个芯片上，合成为中央处理器，简称**CPU**。有程序计数器(PC)、指令寄存器(IR)和控制单元(CU)。一般将运算器和控制器集成到同一个芯片上，称为中央处理器(CPU)。CPU和主存储器共同构成主机，而除主机外的其他硬件装置(外存、I/O设备等)统称为外部设备，简称外设。
- **输出设备（Output Devices）**：将处理的结果展示给用户。输入设备允许用户与计算机交互，例如键盘、鼠标、触摸屏、扫描器、麦克风等。
- **输入设备（Input Devices）**：允许用户输入数据到计算机系统中。输出设备用于将信息传递给用户，如显示器、打印机、扬声器等。

计算机软件的分类

系统软件和应用软件

计算机软件，一般分为系统软件和应用软件

- **系统软件**包括 操作系统，数据库管理系统，语言处理系统(比如编译器)，分布式软件系统，网络软件系统，标准库系统，服务性系统(比如连接程序)。
- **应用软件**包括各种科学计算类程序，工程设计类程序，数据统计与处理程序。

注意：数据库管理系统和数据库系统是有区别的。数据库管理系统是系统软件。而数据库系统一般是由数据库，数据库管理系统，数据库管理员和应用系统构成。所以只能说它里面有系统软件，但并不能说它为系统软件。

三个级别的语言

1)**机器语言**。又称**二进制代码语言**，需要编程人员记忆每条指令的二进制编码。机器语言是计算机唯一可以直接识别和执行的语言。

2)**汇编语言**。**汇编语言用英文单词或其缩写代替二进制的指令代码**，更容易为人们记忆和理解。使用汇编语言编辑的程序，必须经过一个称为汇编程序的系统软件的翻译，将其转换为计算机的机器语言后，才能在计算机的硬件系统上执行。

3)**高级语言**。**高级语言(如C、C++、Java等)是为方便程序设计人员写出解决问题的处理方案和解题过程的程序**。通常高级语言需要经过编译程序编译成汇编语言程序，然后经过汇编操作得到机器语言程序，或直接由高级语言程序翻译成机器语言程序。

由高级语言转换到汇编语言的过程叫做**编译**，由汇编语言转换到机器语言的过程叫做**汇编**，边翻译边执行的叫做**解析**。

机器语言是唯一可以控制cpu的语言，因为它的符号不利于人识别和书写，为了方便理解和记忆，将机器语言换一些通俗易懂的符号，这就变成了汇编语言。一般来说在在编译器中高级语言先转换为汇编在转换为机器语言，也有直接转换为机器语言的情况。

机器语言是计算机唯一可以直接执行的语言，汇编语言用助记符编写，以便记忆。而正则语言是编译原理中符合正则文法的语言。

解释程序的特点是翻译一句执行一句，边翻译边执行：由高级语言转化为汇编语言的过程称为编译，把汇编语言源程序翻译成机器语言程序的过程称为汇编。

- **翻译程序、汇编程序、编译程序、解释程序有什么差别？各自的特性是什么？**

翻译程序是指把高级语言源程序翻译成机器语言程序（目标代码）的软件。

翻译程序有两种：一种是编译程序，它将高级语言源程序一次全部翻译成目标程序，每次执行程序时，只需执行目标程序，因此只要源程序不变，就无须重新翻译，请注意同一种高级语言在不同体系结构下，编译成目标程序是不一样的，目标程序与体系结构相关，但仍不是计算机硬件能够直接执行的程序。另一种是解释程序，它将源程序的一条语句翻译成对应的机器目标代码，并立即执行，然后翻译下一条源程序语句并执行，直至所有源程序语句全部被翻译并执行完。所以解释程序的执行过程是翻译一句执行一句，并且不会生成目标程序。

汇编程序也是一种语言翻译程序，它把汇编语言源程序翻译为机器语言程序。汇编语言是种面向机器的低级语言，是机器语言的符号表示，与机器语言一一对应。

编译程序与汇编程序的区别：若源语言是诸如C、C++、Java等“高级语言”，而目标语言是诸如汇编语言或机器语言之类的“低级语言”，则这样的—个翻译程序称为编译程序。若源语言是汇编语言，而目标语言是机器语言，则这样的—个翻译程序称为汇编程序。

计算机性能指标

计算机性能指标是用来衡量计算机系统性能的各种参数，它们反映了计算机在处理任务时的效率和能力。以下是一些常见的计算机性能指标：

1. 处理器速度 (CPU Speed)

- **时钟速度 (Clock Speed)**：以赫兹 (Hz) 计，通常以兆赫兹 (MHz) 或千兆赫兹 (GHz) 表示。时钟速度越快，处理器每秒能执行的指令次数就越多。
- **核心数 (Core Count)**：多核处理器可以同时执行多个进程或线程，提高多任务处理能力。
- **指令每周期 (Instructions Per Cycle, IPC)**：每个时钟周期内处理器能执行的指令数。

2. 内存 (Memory)

- **内存大小**：以字节 (Bytes) 计，通常以兆字节 (MB) 或千兆字节 (GB) 表示。内存越大，可以同时处理的数据和程序就越多。
- **内存带宽**：内存传输数据的速率，通常以每秒千兆字节 (GB/s) 表示。
- **访问延迟 (Latency)**：CPU从内存读取数据所需的时间。

3. 存储 (Storage)

- **读写速度**：硬盘或固态硬盘读取和写入数据的速度。
- **存储容量**：可以存储的数据总量。
- **输入/输出操作每秒 (IOPS)**：存储设备每秒可以完成的读写操作次数。

4. 带宽 (Bandwidth)

- **网络带宽**：数据通过网络传输的速率，通常以每秒兆位 (Mbps) 或千兆位 (Gbps) 表示。

5. 吞吐量 (Throughput)

- **系统吞吐量**：系统在单位时间内能处理的数据量。

6. 延迟 (Latency)

- **网络延迟**：数据从源头到目的地所需的时间。
- **磁盘延迟**：磁盘完成读写操作的平均时间。

7. 并行处理能力

- **并发线程数**：系统可以同时执行的线程数。

8. 性能基准 (Benchmarks)

- 特定程序或测试套件，如SPEC CPU、PassMark、3DMark等，用于全面评估和比较不同系统和组件的性能。

9. 性能/价格比

- **性价比**：性能与成本的比率，用于衡量购买或升级硬件的经济效益。

10. 可靠性、可用性和服务性 (RAS)

- **故障时间间隔 (MTBF)**：平均故障时间间隔。
- **修复时间 (MTTR)**：平均修复时间。

11. 能效

- **功耗**：设备运行所需的电力，通常以瓦特 (W) 表示。
- **每瓦性能**：单位功耗下的计算能力。