计算机基础

魏华祎

湘潭大学 • 数学与计算科学学院

September 16, 2021

目录

Outline

什么是算法 (algorithm)?

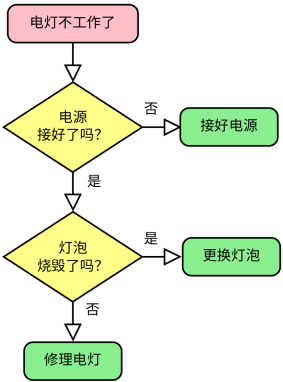

Figure: 应对灯泡不亮的简单流程, 图片来自 Wiki

什么是好的算法 (algorithm)?

- 有可靠的理论分析.
- 有良好的计算复杂性 (时间和空间).
- 易于在计算机上实现.
- 要有具体的编程试验来证明它是行之有效的.

Remark

数据科学与大数据技术这个专业, 是数学和计算机科学的交叉科学, 要学好它需要对计算机这个工具有比较深入的了解.

Outline

中央处理单元 (Central Processing Unit, CPU)

现在的单颗 CPU 通常由多个**核心 (Core)** 组成.

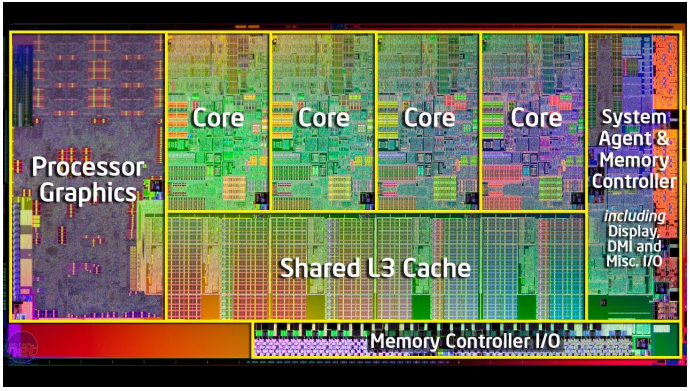

Figure: 多核 CPU.

中央处理单元 (Central Processing Unit, CPU)

现在的单颗 CPU 通常由多个**核心 (Core)** 组成.

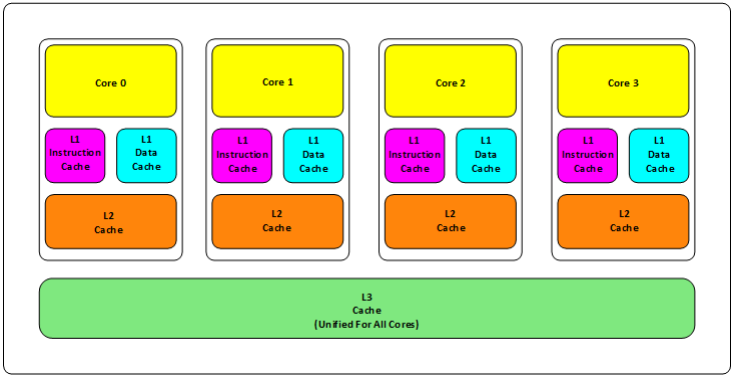

Figure: 多核 CPU 示意图.

中央处理器单元 (Central Processing Unit, CPU)

每个核心主要由**控制器**、**运算器**和**寄存器**等部件组成.

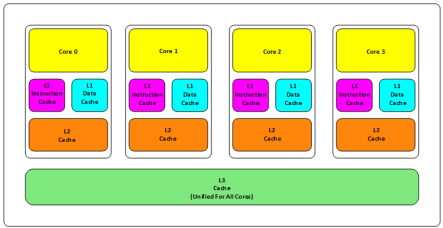

Figure: 多核 CPU 示意图.

控制器是指挥计算机的各个部件按照指令的功能要求协调工作的部件, 是计算机的神经中枢和指挥中心.

中央处理器单元 (Central Processing Unit, CPU)

每个核心主要由**控制器**、**运算器**和**寄存器**等部件组成.

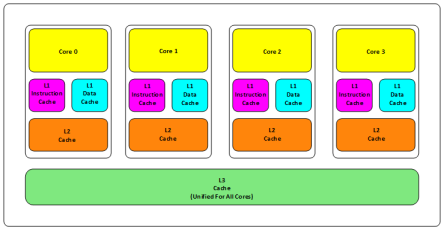

Figure: 现代的多核 CPU 示意图.

运算器 接受控制器的命令,负责完成数据的加工处理任务,其中算术逻辑单元 (ALU) 是其核心部件.

中央处理器单元 (Central Processing Unit, CPU)

每个核心主要由**控制器**、**运算器**和**寄存器**等部件组成.

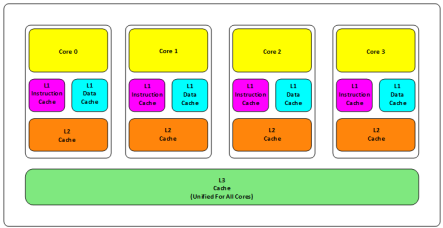

Figure: 现代的多核 CPU 示意图.

寄存器是 CPU 用来存放数据的一些小型存储区域,用来暂时存放参与运算的数据和运算结果, 拥有非常高的读写速度.

中央处理器单元 (Central Processing Unit, CPU)

CPU 另一种重要的部件是**高速缓冲存储器 (Cache)**. 它是主内存中指令和数据进入 CPU 的中转地, 并且 CPU 访问 Cache 的速度远高于内存.

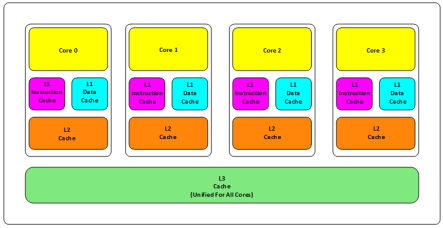

Figure: 现代的多核 CPU 示意图.

Remark

CPU 中引入 Cache 的目的是为了提高系统的效率! 为什么?

中央处理器单元 (Central Processing Unit, CPU)

很多 CPU 的核心都支持**单指令多数据流 (SIMD)**，提供**指令级的并行功能**，即单个指令可以同时操作多组数据。

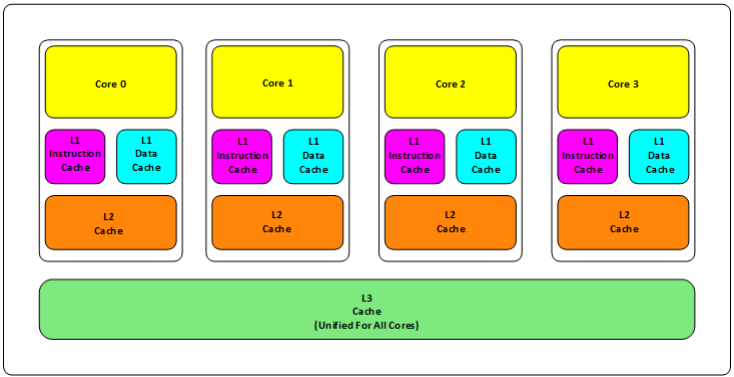

Figure: 现代的多核 CPU 示意图.

CPU 主频

- CPU 在每秒内发出的时钟脉冲信号的个数，即称为 CPU 的主频。
- 常用的单位是 GHZ，如主频为 4.0 GHZ 的 CPU，每秒可产生 40 亿个时钟脉冲信号。
- 通常主频越高的 CPU，单位时间内执行程序指令的速度越快。

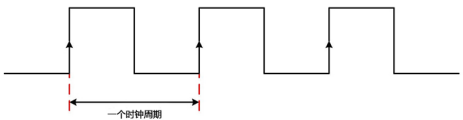

Figure: CPU 时钟信号。

Outline

内存

计算机内存是由很多微型开关组成的存储数据的硬件，其中每个开关的开合状态分别代表 0 和 1。

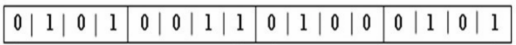

Figure: 计算机内存示意图。

内存

- 内存中的每个开关的状态称为一个**位 (bit)**, 是计算机中内存的最基本单位。
- 八个位组成一个**字节 (Byte)**。
- 为了获取内存中存储的指令和数据,需要对内存中的每个**字节** 进行编号,这个编号就是**物理内存地址**。

$$1KB = 1024B = 2^{10}Byte$$
$$1MB = 1024KB = 2^{20}Byte$$
$$1GB = 1024MB = 2^{30}Byte$$
$$1TB = 1024GB = 2^{40}Byte$$

- **计算机内存的结构是线性的**,可以看成由很多可以存储 0 或 1 小格子组成的长条形结构。

内存

- 32 位的计算机，用 32 位的二进制数表示内存的物理地址，是大可支持 $2^{32}Byte = 2^2 \times 2^{30}Byte = 4GB$。
- 64 位的计算机，用 64 位的二进制数表示内存的物理地址，最大可支持 $2^{64}Byte = 2^{34}GB$。

Outline

简介

操作系统 (OS) 是我们操作计算机硬件的帮手,而且很多时候我们还需要机算机同时处理多个任务,每个任务又可能有多个执行流程,但问题是计算机可利用的计算和存储等资源都是**有限的**。

所以操作系统的主要功能是:在硬件资源有限的条件下,实现多任务的并发运行,尽量提高硬件资源的利用效率,提供资源访问的冲突协调机制。

进程

进程: 是 OS 进行计算机资源分配的最小单位。

- 是 OS 对正在运行的程序的一种抽象。
- 是应用程序的执行实例,每个进程是由私有的虚拟地址空间、代码、数据和其它各种系统资源组成。

线程

线程:是 CPU 核心调度和分配的基本单位。

- 每个进程至少有一个主执行线程,主执行线程终止了,进程也就随之终止。
- 除了进行主线程之外,一个进程还可创建更多的线程。
- 一个进程的所有的线程共享访问进程拥有的资源,但不能直接访问其它进程拥有的资源。
- 线程是用**逻辑内存地址**来访问进程的虚拟地址空间。
- 操作系统为每个进程内的线程提供资源的协调访问机制,如互斥锁(Mutex)和信号量 (Semaphore)。

总结

- 计算机的计算和存储资源是有限的。
- 计算机能表示的信息也是有限的。
- 程序设计者的任务是尽可能高效利用有限的计算和存储资源，完成更多的任务的处理。

Outline

两种操作计算机的方式：CLI 和 GUI

- 命令行界面(Command Line Interface, CLI)
- 图形用户界面 (Graphical User Interface, GUI)
- 大部分的操作系统和应用程序都提供这两种界面。

Remark (理解操作计算机的本质)

- 操作计算机的过程，就是告诉计算机我要执行某个程序，完成某个任务。
- CLI 是通过文本传递信息给操作系统。
- GUI 是通过鼠标放到屏幕的某个区域上，点击发出信息给操作系统。
- CLI 和 GUI 各自对应的进程收到这些信息后，进行解释处理，变成操作系统可以理解的形式传递给操作系统，**操作系统再在磁盘上找到相应的程序，加载到内存中由 CPU 执行，最后返回执行的结果给用户。**

操作系统如何找在磁盘上找东西？

操作系统都有自己的文件系统，文本和二进制(程序和库)等文件都存放在这个系统中。操作系统有两种方式找到这些文件

- **你告诉它**：这种方式需要你记性好
- **它自己找**：这种方式需要它会找(注意它可没记性)
  - 环境变量
  - 系统默认搜索路径

操作系统如何找在磁盘上找东西？

操作系统都有自己的文件系统，文本和二进制(程序和库)等文件都存放在这个系统中。操作系统有两种方式找到这些文件

- **你告诉它**：这种方式需要你记性好
- **它自己找**：这种方式需要它**会**找(注意它可没记性)
  - 环境变量
  - 系统默认搜索路径

操作系统中的各种“器”

- **操作系统命令行解释器**：解释执行操作系统命令及启动应用程序，如 terminal, bash, cmd，shell, prompt 等。
- **编辑器**：编辑文本文件的应用程序，如 vim、emacs、word、wps 等。
- **编译器**：把代码转化为二进制程序和库的应用程序，用于编译型语言，如 C/C++, Fortran 等。
- **调试器**：用读入二进制程序、交互执行及查看程序运行状态的应用程序，如 gdb 等。
- **Python 解释器**：用来解释 Python 命令的命令行。
- **集成开发环境**：集各种“器”于一身的应用程序，给用户提供一个友好便捷的程序开发环境，如 VS Code, pycharm, spyder 等。

Remark

- 这些“器”都是应用程序。
- 使用集成开发环境，不利于你理解事情运行的本质。

Linux 下的常用命令

- 基本格式:“command [-options] parameter1 ...”
- 常用的 Linux 操作命令:
  - cd:文件目录的遍遍历
  - ls:显示文件夹下文件和子文件夹
  - touch: 创建文件
  - mkdir: 创建文件夹
  - rm: 删除文件和文件夹
  - .....

Outline

简介

Git 是分布式版本控制系统，它可以更好的对代码和文档进行版本管理，并支持多人协作。

- 中文学习文档：https://git-scm.com/book/zh/v2。
- gitlab.com 和 github.com 是两个基于 git 的代码文档托管平台。
- 目前我们团队完全基于 git 来进行代码和文档的协作。