第一章　引言

2021-9-27

课程目标

- 入门课程
  - 初学者对计算有一个系统的认识
  - 为后续课程的学习打下一个良好的基础
- 计算机，一个复杂的机器
  - 理解——如何由简单的元件组成计算机，并能执行计算机语言编写的程序
  - 使用——一种高级计算机语言（C语言）编写比较复杂的程序
  - 并能理解——这些程序是如何在计算机这一复杂机器内部执行的

计算机与计算机系统

计算机

- “现代计算机”，“通用电子数字计算机（General-Purpose Electronic Digital Computer）”
- 世界上第一台通用电子数字计算机
  - 1946年，美国宾夕法尼亚大学，ENIAC（电子数字积分器和计算器，Electronic Numerical Integrator and Calculator）

通用计算设备

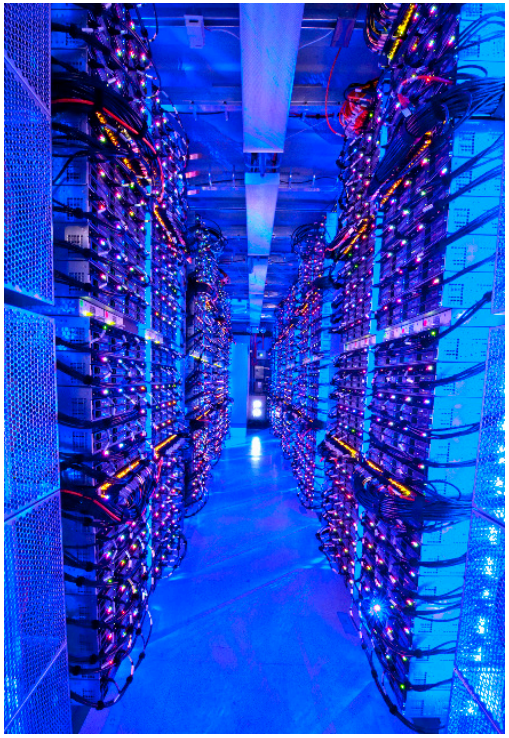
- 通用 对 专用
  - 计算机是一种通用计算设备
    - 阿兰·图灵，1936年发表了一篇论文“论可计算数及其在判定问题中的应用”，给出了通用计算设备的数学描述

通用计算设备思想

- 计算机不是一种专用设备
  - 计算机既可以做加法，也可以做乘法，还可以实现排序或者任何计算
  - 如果想做一种新的计算，不需要重新设计一台新计算机，只需要给它安装合适的软件，就可以达到目的
- 所有的计算机（无论大还是小，快还慢，昂贵还是便宜），如果给予足够的时间和足够的存储器，都可以做相同的计算
  - 所有的计算机都能做几乎完全相同的事情，只是计算速度上有差别
  - 服务器、台式机、PDA等

通用计算设备

=



Personal
Mobile Devices
(PMD)

Cloud
Computing

Alan Turing

- http://www.turing.org.uk/turing/

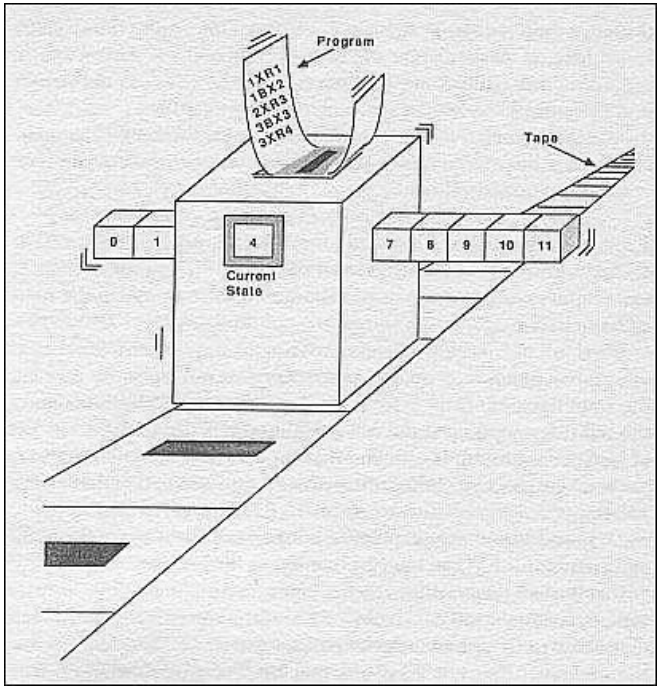
- 1912年6月23日 - 1954年6月7日 ，英国数学家
  - 有限状态自动机/图灵机
  - 人工智能重要的衡量标准“图灵测试”
- 图灵奖
  - ACM于1966年设立，是计算机界最负盛名的奖项，有“计算机界诺贝尔奖”之称。专门奖励那些对计算机事业作出重要贡献的个人
  - 一般每年只奖励一名计算机科学家，只有极少数年度有两名以上在同一方向上做出贡献的科学家同时获奖
  - 目前图灵奖由英特尔公司和Google公司赞助，奖金为250,000美元
  - 获此殊荣的华人是2000年图灵奖得主姚期智

图灵机的基本思想

- 用机器来模拟人们用纸笔进行数学运算的过程——两种简单的动作
  - 在纸上写上或擦除某个符号
  - 把注意力从纸的一个位置移动到另一个位置
- 在每个阶段，要决定下一步的动作，依赖于
  - 此人当前所关注的纸上某个位置的符号和
  - 此人当前思维的状态

图灵机

- 一个抽象的机器
- 有一条无限长的纸带（tape），纸带分成了一个一个的小方格（cell）
- 有一个机器头（head）在纸带上移来移去
- 机器头有一组内部状态（state），还有一些固定的程序（action table）。在每个时刻，机器头都要从当前纸带上读入一个方格信息（symbol），然后结合自己的内部状态查找程序表，根据程序输出信息到纸带方格上，并转换自己的内部状态，然后进行移动。



程序表（action table）

- 程序表，五元组表示
  - <q，b，a，m，q’>
  - q——当前状态
  - b——当前方格原符号
  - a——修改后符号
  - m——机器头移动方向
  - q’——下一状态

示例

- 计算f(x)=x+1
- 纸带原符号，即输入
  - 10100011
- 程序表
  - q_1 0 1 L q_2
  - q_1 1 0 L q_3
  - q_1 b b N q_4
  - q_2 0 0 L q_2
  - q_2 1 1 L q_2
  - q_2 b b N q_4
  - q_3 0 1 L q_2
  - q_3 1 0 L q_3
  - q_3 b b N q_4

从最右边一个格子，状态q_1开始

计算过程

- 10100010 (q_1 1 0 L q_3)
- 10100000 (q_3 1 0 L q_3)
- 10100100 (q_3 0 1 L q_2)
- 10100100 (q_2 0 0 L q_2)
- 10100100 (q_2 0 0 L q_2)
- 10100100 (q_2 1 1 L q_2)
- 10100100 (q_2 0 0 L q_2)
- 10100100 (q_2 1 1 L q_2)

小练习

通用图灵机

- 计算
  - 运用事先规定的规则（action table），将一组数值变换为另一组数值的过程
  - 如果能找到一组确定的规则，按照这组规则，就可以在有限步骤内求出结果——可计算问题
- 图灵机：不是通用的
- 通用图灵机
  - 将action table和输入都写在纸带上

电子设备

- 电子 对 机械
  - “电子”，计算机硬件实现的物理基础
- 计算机是非常复杂的电子设备，计算机执行的计算最终都是通过电子电路中的电流、电位等实现的
- 第七章

数字设备

- 数字 对 模拟
  - “数字”是现代计算机的一种基本特征，也是计算机通用性的一个重要基础
- 在现代计算机里，所有信息都是采用数字化的形式表示
  - 整数、小数、文字、图像、声音等
- 第六章

计算机

- “计算机”，一种能够做计算的机器
- 核心处理部件之一CPU（Central Processing Unit，中央处理器）
  - CPU采用半导体集成电路技术制造，基础材料为硅片，通过复杂的工艺，在只有指甲般大小的硅片上集成了数以亿计的晶体管，被称为“微处理器”
  - 第七章

CPU/程序/指令——第七章

- CPU的工作
  - “指挥信息的处理”，从存储器/内存（memory）里读取下一条指令；
  - “执行信息的实际处理”，执行该指令，即进行加法、乘法等计算工作；
  - 这两项工作循环进行，即读取指令，执行指令，读取，执行……。
- 程序/指令
  - 指令（instruction），计算机执行的一件明确定义的工作
  - 计算机程序（program），由一组指令组成，指令是计算机程序中规定的可执行的最小的工作单位

存储程序控制原理

- 卡片 对 存储器
- 现代计算机的构建思想
  - 美籍匈牙利科学家冯·诺依曼
  - 程序存储在存储器里，CPU负责
    - 指挥信息的处理和执行信息的实际处理
- 第八章

计算机系统

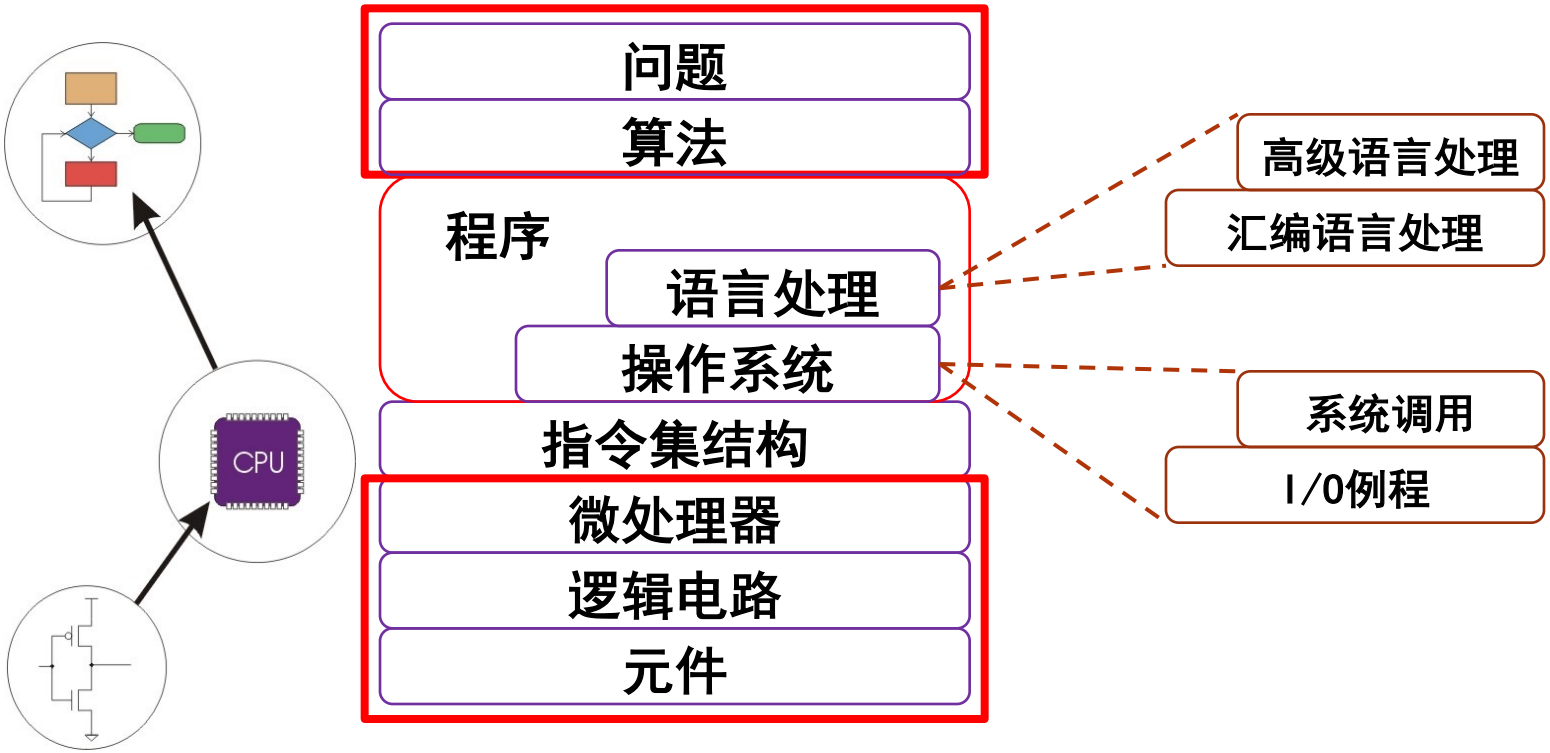
- 计算机系统由硬件（hardware）和软件（software）两部分组成
  - 硬件包括处理器、存储器和外部设备等
  - 软件包括程序和文档
- 在设计硬件或软件时，如果能够同时考虑二者的能力和限制，计算机系统将进入最佳工作状态

计算系统

计算系统

- 人类使用自然语言（即人类所讲的语言）描述问题，计算机则使用电子解决问题
- 将人类的自然语言转换成能够影响电子流动的电压，才能使计算机完成复杂的任务——这种转换是一种有序的系统的转换——计算系统

计算系统的抽象层次


问题
算法
程序
语言处理
操作系统
指令集结构
微处理器
逻辑电路
元件
CPU
高级语言处理
汇编语言处理
系统调用
I/O例程

抽象层次

- 抽象是硬件和软件设计者在解决问题时使用的一种方法
- 每一层次对它的上一层隐藏了自己的技术细节
- 类似：建筑、驾驶汽车…

自顶向下

- 从问题到元件的顺序
- 专业术语
  - 今天可以不理解
  - 学期末回头看时，体会更深

问题

- 人类使用“自然语言”来描述那些希望计算机解决的问题
- 自然语言不能直接作为计算机的指令
  - “歧义性”：为了推断出一句话的含义，听者通常需要根据说话人的发音、语调，语句的上下文来推断
  - 例如：“羽毛球拍卖完了。”
- 计算机是电子设备，只能机械的执行明确的指令，如“Add A, B”是将两个数A和B相加

算法

- 舍弃描述问题的自然语言中的歧义，将自然语言描述的问题转换成一个无歧义的操作步骤，即算法（algorithm）
- 算法是一个逐步计算的过程，该过程一定能够结束，而且每个步骤都能够被明确描述，并能被计算机所执行
  - “有限性”（finiteness）：程序最终能够结束
  - “确定性”（definiteness）：每个步骤都必须是明确的，不应存在歧义性。例如，“A与一个数相加”就是“不确定”的，因为不知道A与哪一个数相加。
  - “有效可计算性”（effective computability）：每个步骤都能被计算机执行。例如，“A除以0”就缺乏可计算性。

算法分析

- 要解决一个问题通常可以采用多种算法，有的算法可能需要的较少的计算时间；有的算法可能需要较少的存储空间
- 算法分析就是对一个算法需要多少计算时间和存储空间作定量的分析
  - 排序算法
  - 查找算法
  - ……
- 后续课程

程序

- 使用程序设计语言把算法转换为程序
- 程序设计语言与自然语言不同，它是用于表达计算机指令的语言，不存在歧义性
- 本课程及后续课程

程序设计语言

- 可以分为高级语言与低级语言两个级别
  - 高级语言和底层计算机有一定的距离，与执行程序的计算机无关，被称为“独立于机器”
  - 低级语言则与执行程序的计算机紧密相关，基本上每种计算机都有自己的低级语言——机器语言和汇编语言
  - 以“将两个数A和B相加”为例，C语言可以表示为“A+B”；而用某种机器的汇编语言表示，可以为“Add A, B”，其机器语言则为“0001001001000000”
  - 本课程：C语言、DLX计算机的机器语言和汇编语言
- 本课程及后续课程

语言处理

- 高级语言程序，必须将其翻译成执行程序作业的机器（目标机器）的指令，即机器语言，才能在目标机器上执行
  - 把高级语言翻译成机器语言的工作通常可以由一个翻译程序来完成
- 对于使用某种机器的汇编语言编写的程序，如果要在该机器上执行，由一个叫做汇编器的程序来完成从汇编语言程序到该机器指令集的翻译工作

操作系统

- 如何把编写的程序输入计算机？如何把计算机执行的结果输出给用户？
- 最初的操作系统包含的就是支持输入/输出操作的设备管理例程
  - 第十二章、第十三章
- 随着技术的发展，操作系统包含了文件管理、内存管理、进程管理等主要功能
- 后续课程

指令集结构

- 将高级语言程序翻译成机器语言，依据就是目标机器的指令集结构（Instruction Set Architecture，ISA）
  - 如果需要将某种高级语言（如C语言）翻译到某种目标机器（如IA-32）上执行，必须使用相应的翻译程序
- 指令集结构是编写的程序和执行程序的底层计算机硬件之间的接口（Interface）的完整定义

指令集结构

- 指令集结构指明了计算机能够执行的指令集，也就是说计算机能够执行的操作和每一步操作所需的数据
  - 1979年，Intel公司设计的IA-32指令集结构
  - 1986年发布的MIPS指令集结构
  - PowerPC（IBM和Motorola），Alpha（COPMPAQ），PA-RISC（HP），SPARC（SUN和HAL）以及最新的IA-64(Intel)等

iPhone

- ARM指令集
  - 和MIPS类似，由Berkeley RISC演变而来
  - 低功耗，且能满足性能
  - ARM公司
    - 自己不制造芯片，只将芯片的设计方案授权（licensing）给其他公司，由他们来生产
    - 软硬件接口

- 第九章：DLX指令集结构
  - MIPS的简化版
- 第十章：DLX机器语言
- 第十一章：DLX汇编语言
  - 汇编语言处理
  - 高级语言处理：C到DLX汇编语言

微处理器

- 每一种指令集结构都可以采用多种微结构来实现
- 对于计算机设计者来说，每一种实现都是一次对微处理器的成本和性能之间的平衡
- 第九章

微处理器

- IA-32指令集结构
  - 从1985年Intel实现的80386微处理器，之后的80486、80586微处理器，到1998年推出的Pentium（奔腾）微处理器，都是采用不同微结构对IA-32指令集结构的实现
- MIPS指令集结构
  - Cisco、Nintendo、Sony和SGI等公司生产，实现了不同的微处理器，用于Sony、Nintendo的游戏机，Cisco的路由器和SGI超级计算机中
- 后续课程

逻辑电路

- 组成微处理器的每一个组件的逻辑电路也有很多选择，因为设计者也需要考虑如何尽量平衡成本和性能
  - 例如，对于组成微处理器的加法器的实现，选择超前进位逻辑电路，比选择行波进位电路，计算速度更高
- 第七章、后续课程

元件

- 每一种基本的逻辑电路都是由特定的物理元件实现的
  - CMOS（Complementary Metal Oxide Semiconductor，互补金属氧化物半导体）逻辑电路采用金属氧化物半导体晶体管制造
  - 双极型逻辑电路则采用双极型晶体管构成
- 第七章

课程大纲

- 第一部分（第2~5章）
  - 问题
  - 算法
  - 程序：高级语言
    - C语言
- 第二部分，自底向上（第6~14章）
  - 元件
    - CMOS
  - 逻辑电路

课程大纲

- 微处理器
- 指令集结构
- 程序
  - 机器语言、汇编语言
  - 语言处理：汇编语言处理，高级语言处理
  - 操作系统：I/O例程，系统调用

DLX

- 第三部分（第15~17章）
  - 程序
    - C语言深入主题
    - 高级语言处理

DLX

- DLX指令集是基于美国加州大学伯克利分校计算机系Patterson教授和斯坦福大学计算机系Hennessy教授在《计算机系统结构：一种定量的方法（第二版）》一书中给出的DLX指令集，并根据需要对其进行了裁减和扩充。
- DLX不是一个真实的计算机，但是其设计方法覆盖了现代真实机器实现的一般方法，而且由于其简化的特性，使其能够被初学者所理解。
- 基于FPGA给出了实现

本章小结

- 通过“逐层转换”，C语言程序通过电子的流动得以实现
- 在解决实际问题时，不必陷入“逐层转换”的细节之中
  - 当设计一个复杂的计算机应用程序时，如设计一个文字处理软件，不需要考虑以上9个抽象层次的每一处细节，而应该集中考虑问题的本质，更多地关注从问题到算法，从算法到程序的转换
  - 在使用门电路设计逻辑电路时，也不必考虑每个门电路的内部结构
- 但是，如果文字处理软件或逻辑电路在工作时出现问题，此时，就需要了解这些细节
  - 逻辑电路出现的问题可能与门电路的某种内部结构有关，因此，为了更好的解决问题，则必须了解“逐层转换”的过程

习题

- 1.2
- 1.6(使用自然语言描述即可)
- 1.7
- 1.8
- 1.9