

# 第1章 认识计算机



## 1.1 计算思维概述

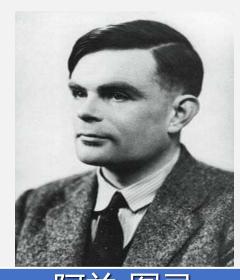


## 计 算 思 维

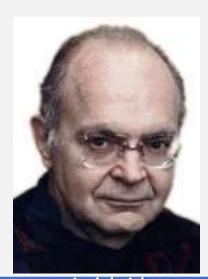
计算设备的应用与发展是人类社会进步的必然产物——计算问题不处不在、而计算机也将与时俱进



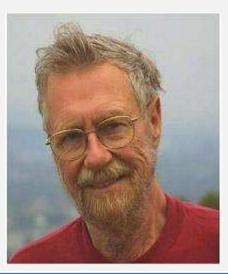
### 20世纪, 计算机科学界涌现出了一批天才。



阿兰·图灵 (Alan Turing)



唐纳德 (Donald Knuth)



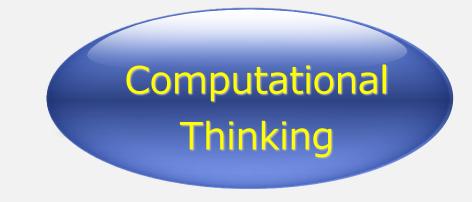
艾兹赫尔·戴克斯特拉 (Edsger Dijkstra)

他们建立了改变人类生活的现代计算机工业,影响了整个世界。



### 计算思维的概念

• Edsger\_Dijkstra 我们所使用的工具影响着我们的思维方式和思维 习惯,从而也将深刻的影响着我们的思维能力。



#### 周以真

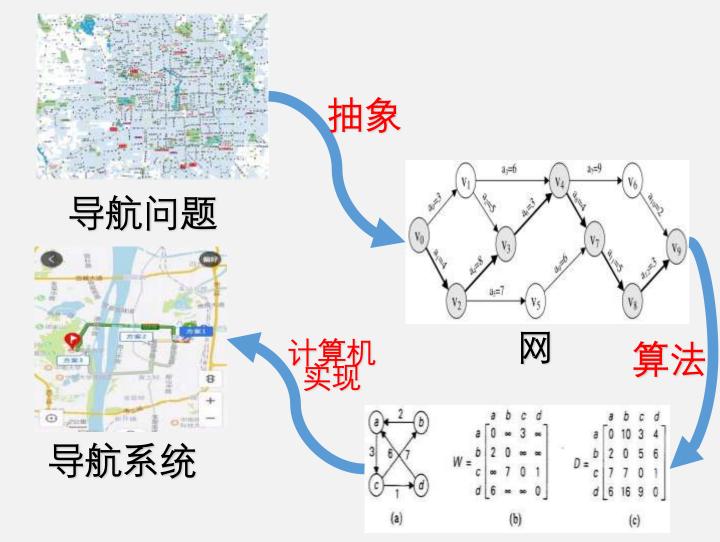
计算思维是运用计算机科学的基础概念进行问题求解、系统设计以及人类行为理解等涵盖计算机科学之广度的一系列思维活动。

 Computational thinking will be a fundamental skill used by everyone in the world by the middle of the 21st Century.



### 理解计算思维

- •用四个字来概括:
  - 抽象、算法
- •用八个字来概括:
  - 合理抽象、高效算法



计算机的出现为计算思维的实现提供了高效的手段

最优路径



- 概念化,不是程序化。
- 基础的,不是机械的技能。
- 人的,不是计算机的思维。
- 数学和工程思维的互补与融合。
- 面向所有的人,所有地方。
- 是思想,不是人造品。
- 计算思维最根本的内容: 抽象(Abstraction)和自动化 (Automation)



### 计算思维能力

### 计算思维能力的概念

面对一个新问题,运用所有资源将其解决的能力。计算思维能力的核心是问题求解的能力。

- ❤发现问题
- ♥寻求解决问题的思路
- ✔分析比较不同的方案
- ♥验证方案



### 计算思维能力的培养

- 深入掌握计算机解决问题的思路, 更好地用好计算机。
- 把计算机处理问题的方法用于各个领域,推动在各个领域中运用计算思维,更好地与信息技术相结合。
- 求解能力是"教"不会,"学"不会的,只能"练"会。



### 计算机与计算思维的关系

- 计算机能干什么?
  - 存储更多的数据---大数据
  - 解决更复杂的问题---人工智能。
  - 交流更方便些-----听说。
  - 以亿亿次/S的速度, 模拟无法实现或耗资巨大的过程等。

### 计算机不能干什么?

人机分界在于"思考"二字。计算机所不具备:直觉、综合、机敏、 灵感。由人来创造性地研究各种所需的算法、模型、方法。

计算机是工具,帮助提升人的能力 计算思维是人类思维活动,有助于计算机解决更为复杂的问题。



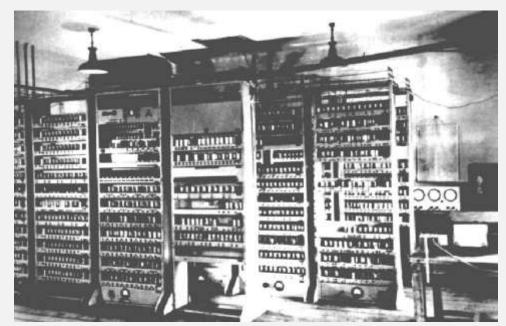
## 1.2 冯. 诺依曼体系结构



1945年,冯·诺依曼提出了计算机共同遵守的基本规则,被称为"冯·诺依曼体系结构"。



冯•诺依曼



1949年 EDSAC

#### ○五大基本部件;

〇采用二进制数表示指令和数据;

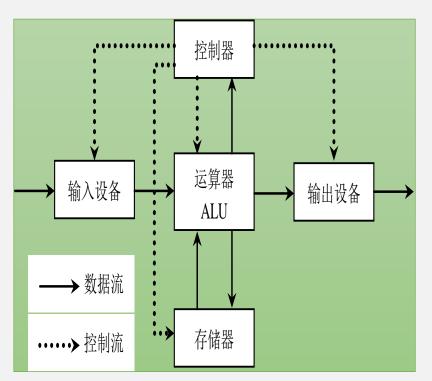
○存储程序原理

六十多年来,计算机技术突飞猛进,但计算机系统基本结构没有变



### 五大构成部件

典型的冯·诺依曼计算机组成---运算器为核心



输入设备:程序和数据送至计算机中;

运算器: 完成各种算术、逻辑运算;

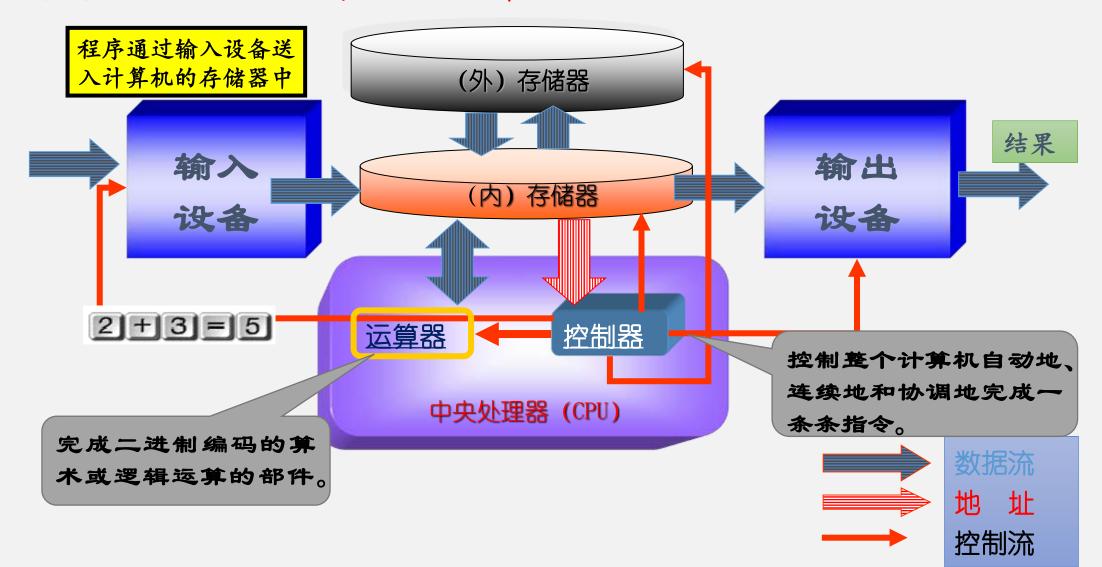
存储器:长期记忆程序、数据、结果的能力;

输出设备:按照要求将处理结果输出给用户。

控制器:根据需要控制程序走向,并能根据指令控制机器的各部件协调操作:



### 现代计算机组成---以存储器为中心

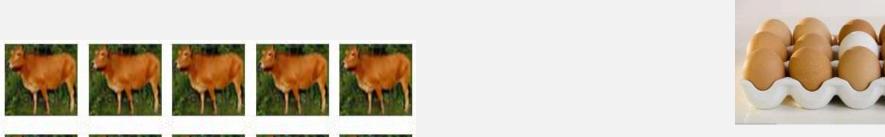




### 二进制存储

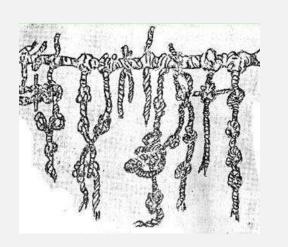
进制仅是一种计数策略---何种进制都有存在的价值











二进制: 1111

八进制: 17

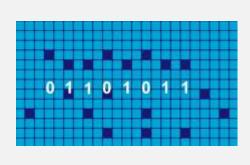
十进制: 15

十六进制: F



### 计算机采用二进制





简单: 技术实现、运算控制

可靠: 抗干扰



转换为二进制



100+1000 1100 转换为R进制



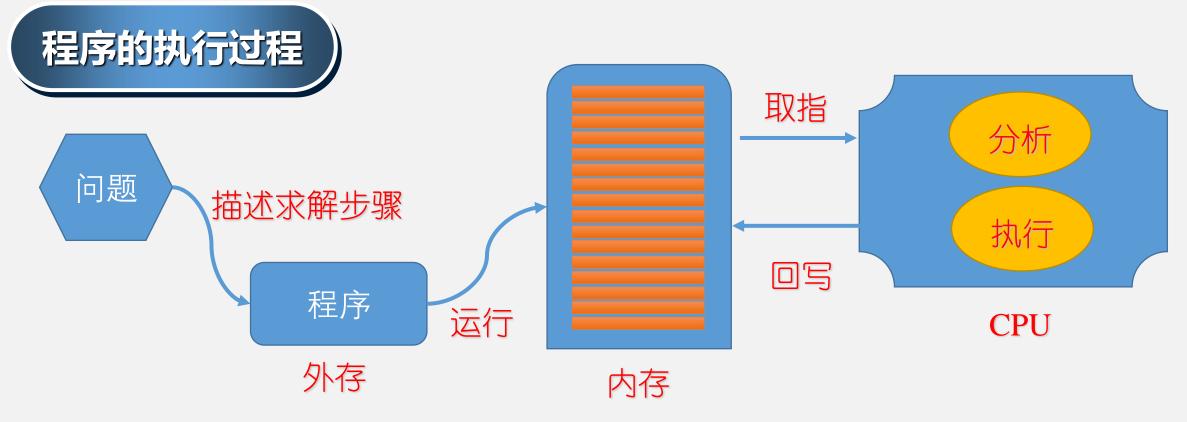
R进制显示 12

输入R进制 4+8



存储程序原理

计算机的工作原理: "存储程序" + "程序控制"



无需干预 自动完成



## 指令与程序



指令是对计算机进行程序控制的最小单位。 计算机所识别的指令集合称为计算机的指令系统。





例如: 10110000 00000101

这是一条2字节指令,第1个字节(即10110000)表示操作码,第2个字节(即00000101)表示操作数。

含义:把数5送入累加器A。



# 指令系统

指令系统: CISC (20世纪90年代前)、RISC (ARM)。不同的CPU架构,采用不同的指令集。

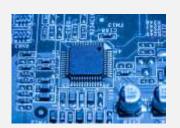


## 程序

完成一项特定任务的一组指令序列。

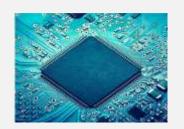
源程序(高级语言书写)

编译程序A



处理器A-指令系统A

编译程序B



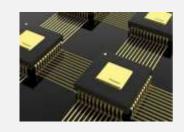
处理器B-指令系统B

编译程序•••••



• •

编译程序N



处理器N-指令系统N



# 1.3 计算机的硬件组成



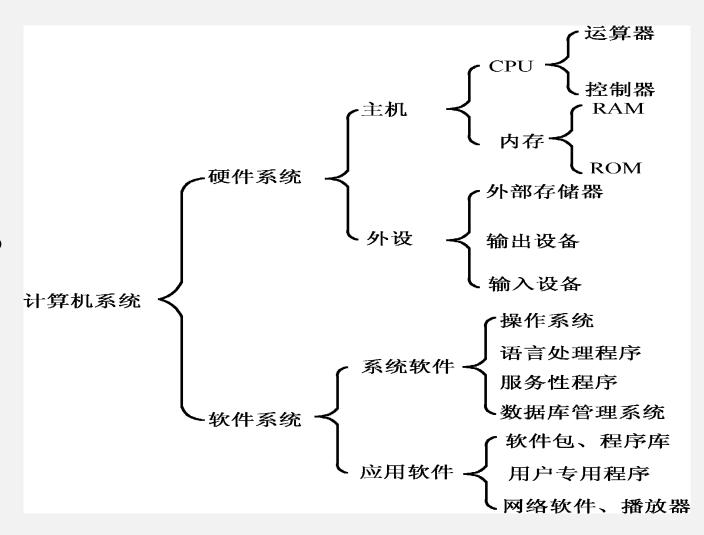
### 硬件与软件

•硬件: 物理实体, 决定性能;

•软件:逻辑实体,决定功能。

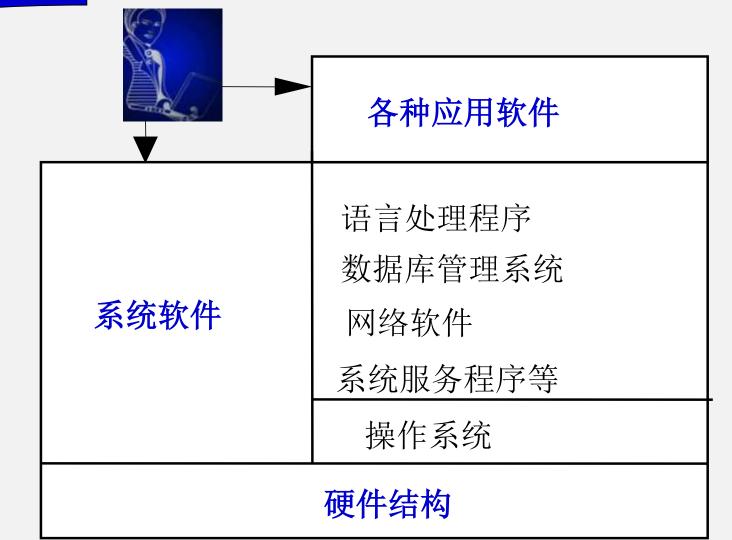
•相互依赖协同工作

•相互促进协同发展





### 软硬件层次关系





### 部件与配件

### 5大部件

- 输入设备
- 输出设备
- 存储器
- 运算器
- 控制器



















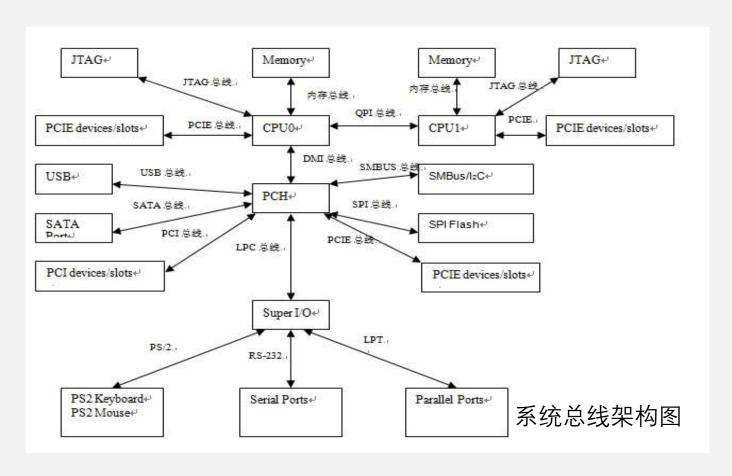




# 总线

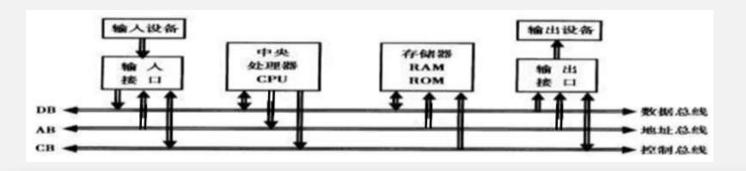
总线是计算机各种功能部件之间传送信息的公共通信干线。

主机的各个部件通过总线相 连接,外部设备通过相应的 接口电路与总线相连接,形 成了计算机硬件系统。





#### 依据传输的信息种类



地址总线

地址总线AB用于传送地址信息:读写位置

单向:只能从CPU传向存储器或I/O端口

分类

数据总线DB用于传送数据信息

数据总线

双向: 可把CPU的数据传送到存储器或输入输出接口等其它部件, 也可将其它部件的数据传送到CPU

控制总线

控制总线主要用来传送控制信号和时序信号 读写信号,中断请求、回答与响应信号,工作状态信号以及其他各种功能控制信号。



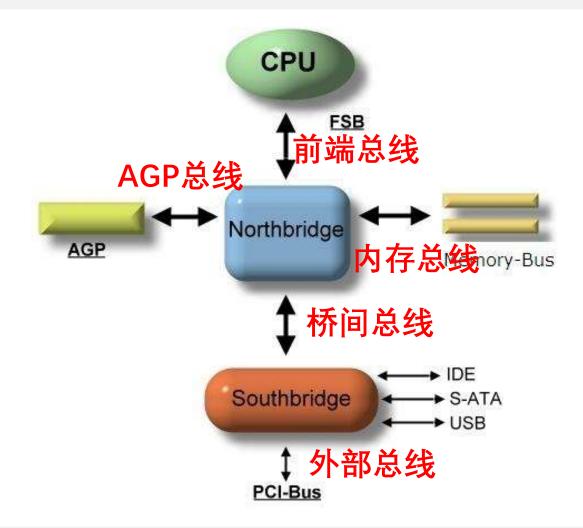
### X86架构 CPU总线

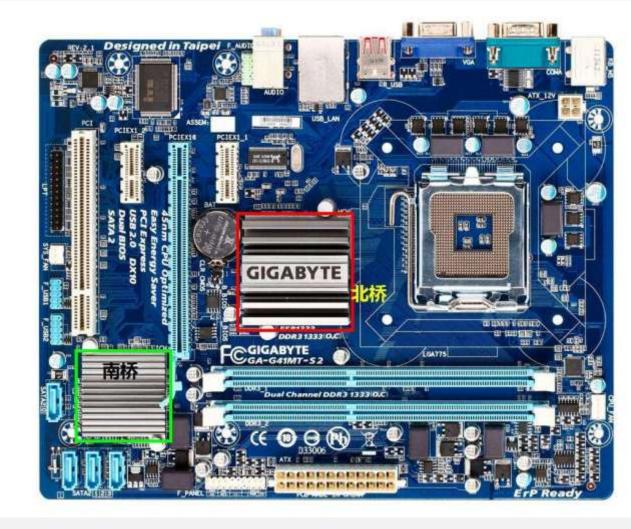
Intel从8086开始, 286、386、486、586、P1、P2、P3、P4都用的同一种CPU 架构, 统称X86。



早期PC, CPU/RAM/IO都是在一条总线上, 所有部件必须在同步的模式下工作。







南桥、北桥芯片解决速度不匹配问题。

北桥主要控制 CPU内存显卡等高速设备。南桥芯片负责I/O总线之间的通信。



## 1.4 计算机软件



### 软件的概念及特点

软件

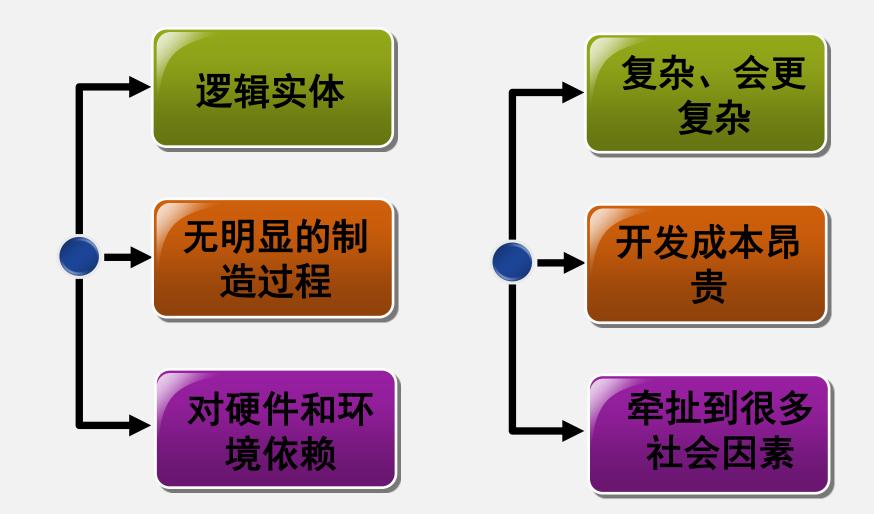
软件: 与计算机系统操作有关的程序和文档。

程序-指完成特定功能的指令序列以及所处理的数据;

文档-与软件开发、维护和使用有关的文字材料,是软件的必要的组成部分。



### 软件的基本特点



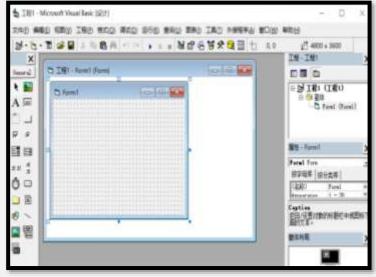


### 软件的分类



指控制和协调计算 机及外部设备,支 持应用软件开发和 运行的系统。





操作系统

语言处理程序

数据库管理系统

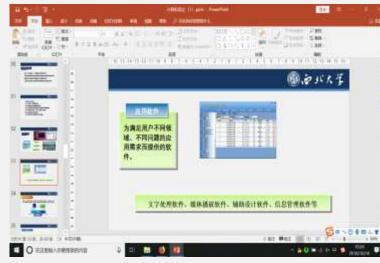




#### 应用软件

为满足用户不同领 域、不同问题的应 用需求而开发的软 件。





文字处理软件、媒体播放软件、辅助设计软件、信息管理软件、导航系统等。





### 软件危机与软件开发模型

软件危机

软件危机是指在计算机软件的开发和维护过程中所遇到的

- 一系列严重问题。包含两方面的问题:
  - (1) 如何开发软件,以满足不断增长,日趋复杂的需求;
  - (2) 如何维护数量不断膨胀的软件产品。



软件危机 主要表现

- 软件开发成本和进度的估计不准确
- 对系统不满意的现象经常发生
- **软件产品的质量往往不可靠**
- **软件的可维护程度非常低**
- 软件的成本不断提高
- **软件开发不能满足硬件和人们需求**



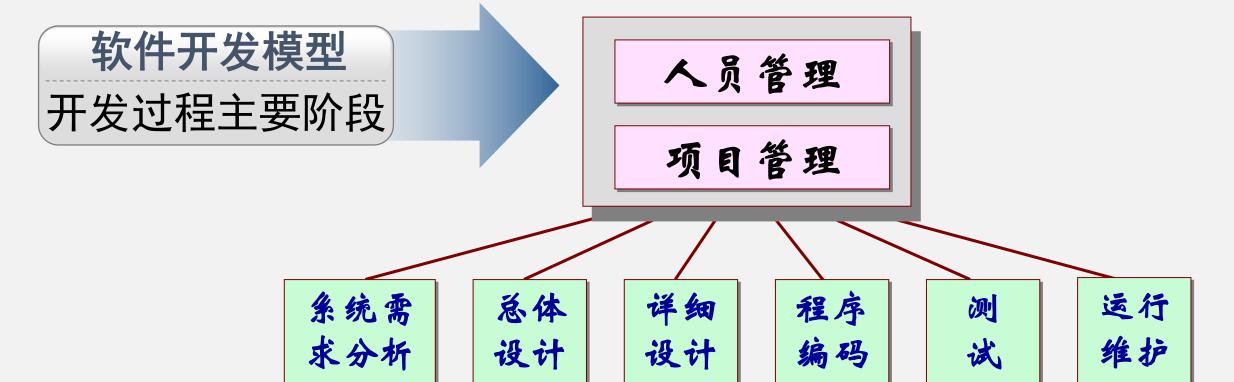
## 软件开发模型

软件开发模型是软件开发的全部过程、活动、任务和管理的结构框架。

它能清晰、直观地表达软件开发全过程,明确规定要完成的主要活动和任务,是软件项目研发的基础。

选择合适的开发模型十分重要。







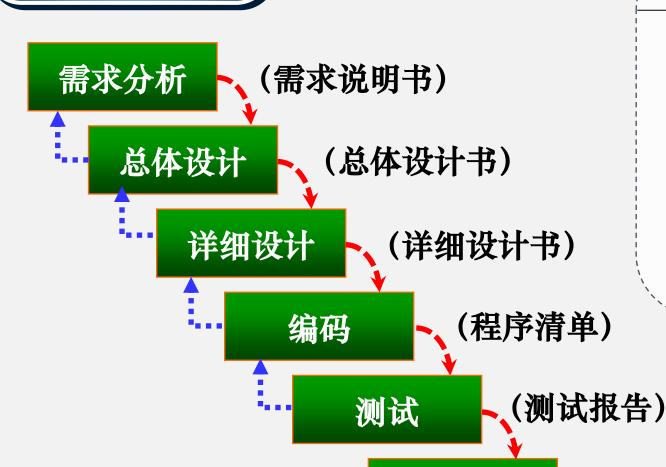
没有任何一种方法能够解决软件危机中的所有问题,所以在软件开发的各个阶段采用综合治理的方法。

软件开发模型直接影响软件开 发的周期和软件质量,是软件 开发的组织管理形式,是软件 工程最重要的内容之一。





## 瀑布模型简介



运行及维护

### 瀑布模型

将软件生存周期所有活动规定 为按线性顺序联接的若干阶段: 需求分析、总体设计、详细设 计、编码、测试、运行和维护。 各阶段由前至后、相互衔接, 如同瀑布流水,逐级下落。

(维护报告,改进的系统)



上一阶段的成果是本阶段的工作对象;

特点

本阶段的成果,作为下一阶段的输入;

任何阶段都要进行工作进行评审,若得到确认,则继续 下阶段的工作,否则返回前一阶段或更前一阶段。

优点

提供了一个开发模板,使得分析、设计、编码、测试、运行维护可以在该模板的指导下应用。

各阶段的质量控制尽可能避免错误的累积。



### 缺点:

- •缺乏灵活性,不能适应用户需求的改变
- •对于需求不能完全确定的软件开发项目将产生很大的风险。

### 通常使用场合:

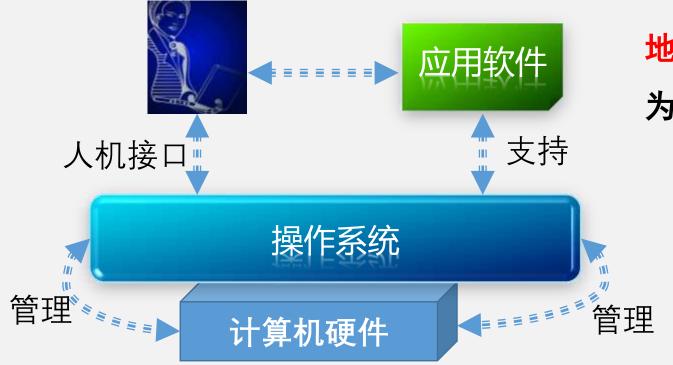
- 需求分析做得比较好的系统
- •二次开发系统



# 1.5 操作系统简介



操作系统简称0S,负责管理和控制计算机系统中的全部软、硬件资源,合理地组织计算机的工作流程,为用户应用程序的运行提供一个友好的界面和良好的工作环境。



地位:是系统硬件之上的第一层软件, 为其他软件提供单向支撑作用。

### 主要设计目标:

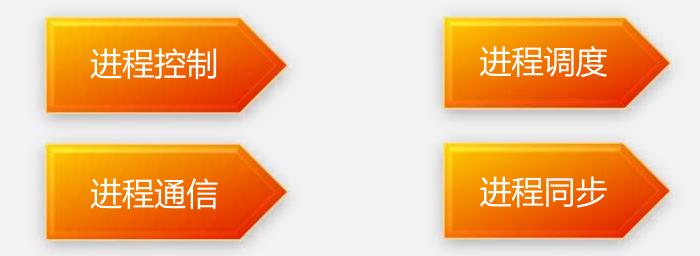
向用户提供方便、简单的使用环境; 提高系统资源的利用。



## 操作系统的基本功能

### 处理机管理

处理机管理的主要任务是对中央处理机的分配和运行实施有效的管理,提高CPU的利用率。





# 存储器管理

程序只有装入主存时才有可能被处理机执行,存储器管理为更多的任务分配所需的存储空间。

内存分配

存储保护

存储扩充



## 设备管理

设备是计算机中的重要资源,设备管理的主要任务是控制设备和CPU之间的I/0操作。

缓冲区管理

设备分配

设备处理

虚拟设备

设备独立性

1/0设备种类繁多,特性和操作方式相差甚大

设备管理成为操作系统中最繁杂且与硬件最紧密的部分。



## 文件管理

文件管理是操作系统中实现文件统一管理的一组软件、被管理的文件以及为实施文件管理所需要的一些数据结构的总称。

从系统角度来看,文件系统是对文件存储器的存储空间进行组织,分配和回收,负责文件的存储,检索,共享和保护。

- (1) 文件存储空间的管理。 (2) 目录管理。
- (3) 文件保护。 (4) 文件操作管理。



## 用户接口

操作系统提供两种形式的接口供用户使用。

(1)命令接口

提供一组命令供用户直接或间接控制自己的作业;

(2) 程序接口

提供一组系统调用供用户应用程序和其他系统程序调用操作系统的功能。



## 常见操作系统



#### 安卓操作系统

Google发布基于Linux平台的开源移动手机平台。

硬件厂商:摩托罗拉、三星、LG、联想,中兴等。

优点:具备触摸屏、高级图形显示和上网功能,界面强大。

缺点:由于跨平台的优势导致了安卓系统的不兼容性;开放性带来的不安全。





### IOS操作系统

最初为iPhone设计,后来用到iPod touch、iPad 以及Apple TV等产品上。

硬件厂商: 苹果公司

优点: 全触摸设计, 第三方软件多。

特点: 系统封闭

ios

IOS产品的封闭性,所有的APP都运行在iPhone, iPad等iOS产品当中,有着很高的硬件利用效率。



#### **Windows Mobile**

Microsoft开发的用于Pocket PC 和Smartphone 的软件平台。 将Windows 桌面扩展到了个人设备中。

硬件厂商: HTC、三星电子、LG、山寨厂商。

优点: 界面和操作都和Windows十分接近, 各种保存在电脑或手机里的信息、资料可以轻松实现共享, 有大量的应用软件可供用户选择。

**缺点**:占用系统资源高、容易系统崩溃。







主流: Windows系统、Linux系统和Mac OS系统。

### Windows系统

微软公司开发的窗口化操作系统。采用GUI 图形化操作模式,使用广泛。





### Mac OS操作系统

苹果计算机公司为它的Macintosh计算机设计的操作系统。最早采用GUI图形用户界面、多媒体应用、鼠标等。





#### Linux

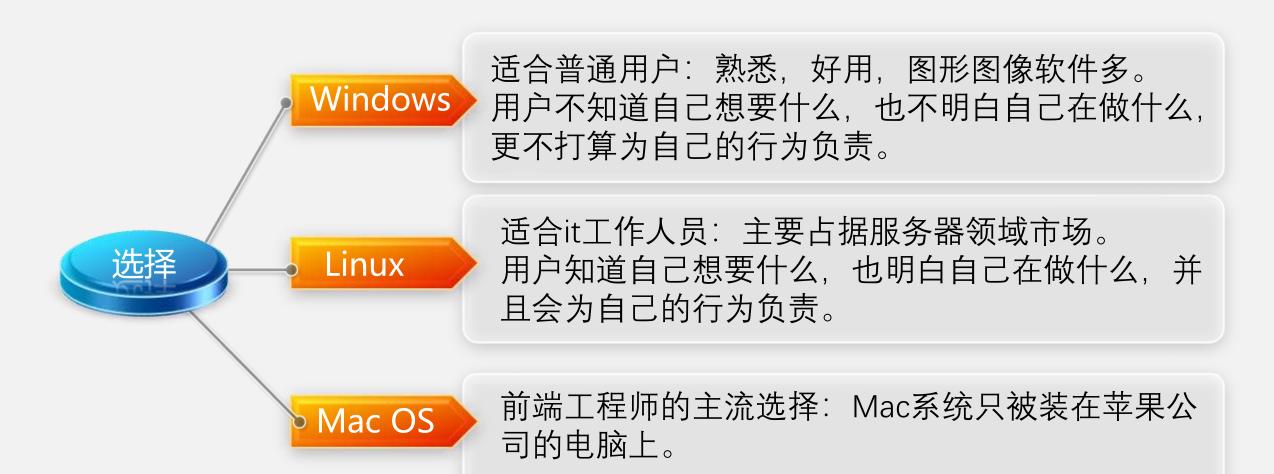
Linux是一种自由和开放源码的操作系统,存在多种不同的Linux版本(都使用Linux内核)。Linux可安装在各种计算机硬件设备中,比如手机、平板电脑、路由器、视频游戏控制台、台式计算机、大型机和超级计算机。

世界上运算最快的10台超级计算机运行的都是 Linux操作系统。











# 1.6 语言处理程序



# 程序设计语言

### 程序设计语言是用于书写计算机程序的一组记号和一组规则。





### 机器语言

- 机器语言是计算机系统唯一能识别的、不需要翻译直接供机器使用的程序设计语言。
- 用机器语言编写程序难度大、直观性差、容易出错,修改、调试也不方便,
- 计算机能够直接识别,程序运行速度最快。

1+2+……100的部分代码



### 汇编语言

- 机器语言助记表示。
- 和机器语言一样,也是面向机器的程序设计语言,通用性差,使用不方便。
- 机器语言和汇编语言一般都称为低级语言。

MOV AX, 20

MOV CX, 100 计算机不识别,需要翻译。

ADD AX, CX



### 高级语言

高级语言的书写方式更接近人们的思维习惯,程序更便于阅读和理解,出错时也容易检查和修改。

MOV AL,0 MOV BL,0 LAB:INC AL ADD BL,AL CMP AL,100 JNE LAB 汇编语言



# 语言处理程序

语言处理程序一般是由汇编程序、编译程序、解释程序和相应的操作程序等组成,其作用是将源程序翻译成计算机能识别的目标程序。



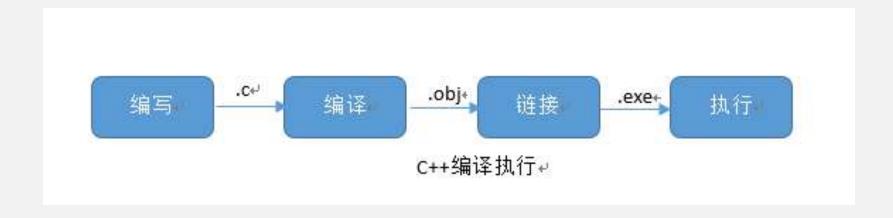
使用机器语言书写的程序,无需翻译。

汇编语言书写的程序无法独立于机器(特定的CPU体系结构),要翻译成机器指令才能执行。

高级语言独立于机器的特性,靠编译器为不同机器生成不同的目标代码。将高级语言编译到什么程度跟编译的技术有关。

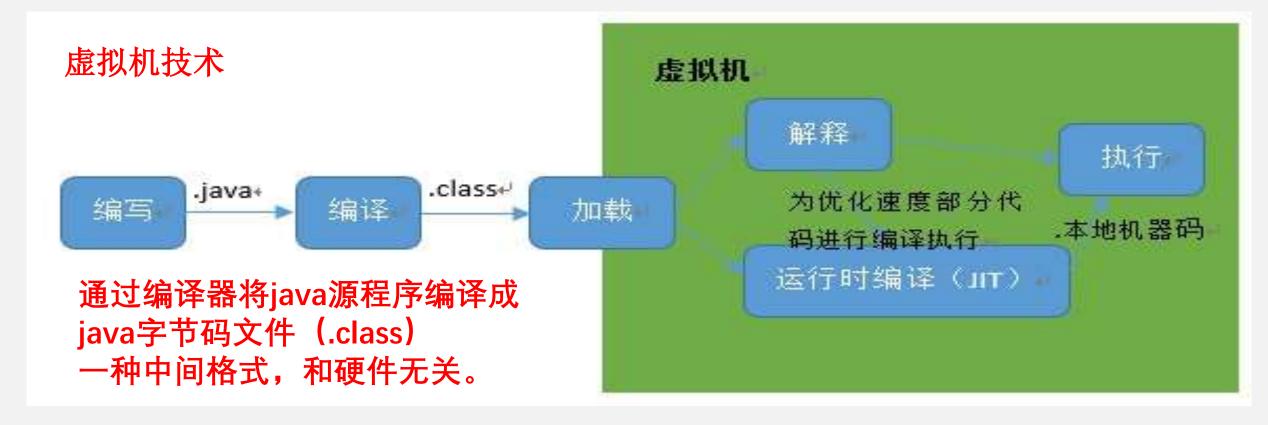


### 解释和编译



生成可执行的EXE文件,指令序列,依赖具体的硬件平台。 **跨平台需要重新编译。** 





虚拟机由软件实现,将字节码文件转换为具体平台可执行的机器指令,实现了JAVA的跨平台性。

JAVA有多种JVM,不同的JVM中有针对不同平台的解释器,可以把相同的字节码文件翻译成相应平台可以识别执行的机器指令码。



# 1.7 无处不在的计算



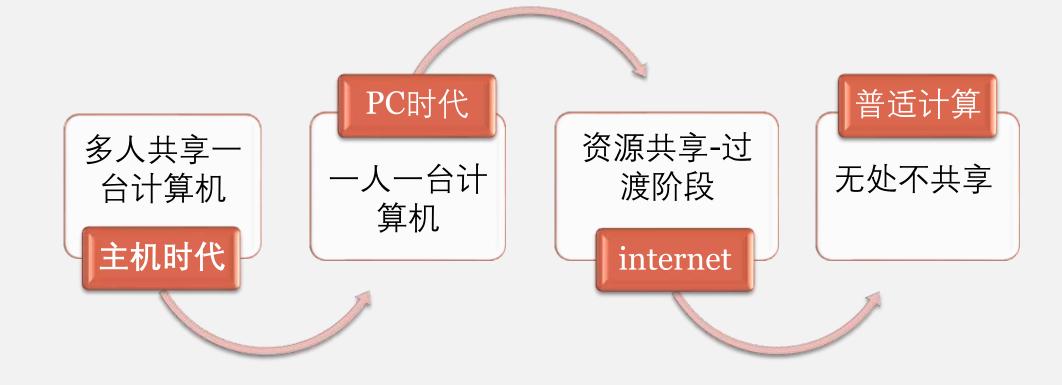
新技术迅速发展: 下一代互联网、云计算、大数据、人工智能、区块链

无处不在的计算硬件正在出现: 无人汽车、手表、眼镜、包包、牙刷、镜子、 冰箱、灯、各种显示屏等。

计算无处不在,连接随手可及



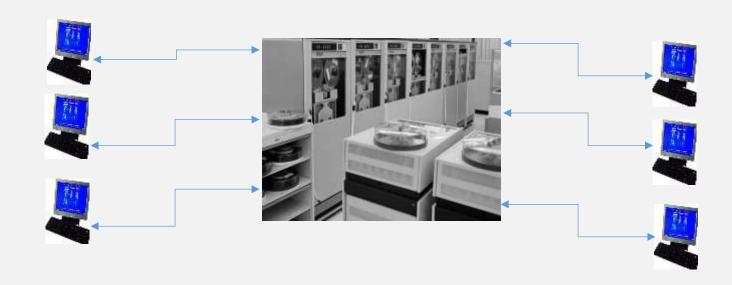
# 计算模式的演变





# 主机时代

**计算机是稀有资源**。一般放在密闭房子里由专家操作,其他 普通用户通过终端共享一台计算机。







个人可以有自己的计算机,利用它完成某项任务,但需要自己操纵。







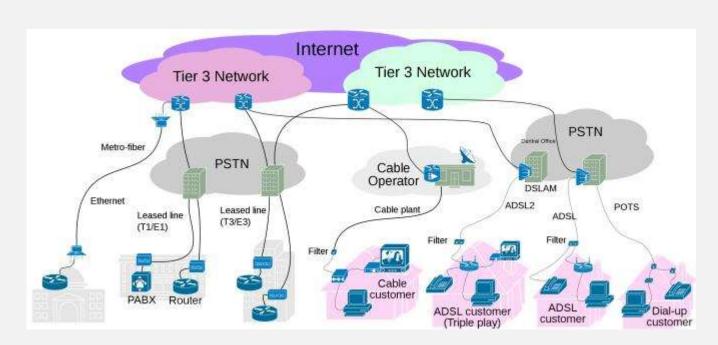


# internet时代

资源共享为目的,海量的用户及其资料会被相互连接,同时也将主机时代和PC 机时代的大量设备连在一起。

大量的信息联网必将会出新的计算关系。



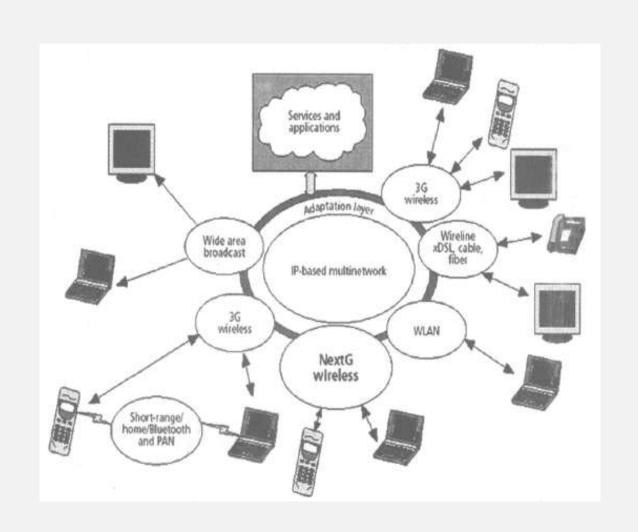




## 普适计算时代

无处不在的计算: 用户可以更为方便的获取自己所需信息。大量的计算设备嵌入在墙壁、椅子、衣服、电灯开关、汽车等一切东西中。

基本特征:深度的嵌入计算,即连接现实世界中一切具有计算能力但规模大小不同的东西。





## 普适计算概述

## 普适计算的概念

把计算机融入到环境中去,使人们关注的重点从操作工具转移到执行任务本身上来,可以在任意时间、使用任意设备、通过任意网络来获得所需的服务。



计算资源的丰富与廉价 互联网(无线网络)的广泛使用

普适计算环境下人的注意力就成为一种稀缺资源。 普适计算关注的是人们的注意力以及人们对计算的满意程度,它是一种以人为中心的计算模式。





# 普适计算的要求

• 普适性

数量众多的计算设备嵌入到环境中,通过这些设备用户可以随时随地得到计算服务。

• 透明性

整个计算系统不断地更新和学习,程对于用户透明。可以使用户最大程度地将注意力放在要完成的任务上。



### •动态性

用户通常处于移动状态:特定的空间内用户集合将不断变化;

移动设备动态地进入退出:计算系统的结构发生动态变化。

•自适应性

计算系统可以感知和推断用户需求,自发地提供用户需要的信息服务。

•永恒性

计算系统不会关机或者重启,计算模块可以根据需求、系统错误或系统升级等情况加入或离开计算系统。



## 普适计算涉及的技术

