

0 计算导论

郑海永

zhenghaiyong@ouc.edu.cn

中国海洋大学 电子工程学院



目录 I

1 计算机的基本原理

- 从数学危机到图灵机
- 图灵机的构成与原理
- 计算机为什么能计算

2 计算机的历史与未来

- 历史上的计算设备
- 从电子管到云计算

3 程序运行的基本原理

- 冯诺伊曼式计算机
- 存储器的种类特点原理类型
- CPU 指令的执行
- 程序的执行

4 操作系统与编程语言

- ## ● 操作系统

1 计算机的基本原理

- 操作系统
- 编程语言

1 计算机的基本原理

- 从数学危机到图灵机
- 图灵机的构成与原理
- 计算机为什么能计算

- 操作系统
- 编程语言

图灵奖



Since 1966

图灵奖



Since 1966

Turing

- 图灵是谁？
- 他做了什么？
- 为什么要以他的名字命名？

- 古希腊毕达哥拉斯学派 VS. 希帕索斯
- “数即万物”（整数或整数之比）VS. 无理数（勾股定理、 $\sqrt{2}$ ）
- 危机缓解：200 年后几何学 危机解决：19 世纪下半叶实数理论

- 1070 · 守恒理论、极限论 新的问题：处处不可微的

- 连续函数 $\Rightarrow$ 直观及几何思考不可靠

- 实数论 $\Rightarrow$ 集合论

第三次数学危机 十九世纪下半叶

- 康托尔集合论 VS. 罗素悖论（理发师悖论）
- 从自然数与康托尔集合论出发可建立起整个数学大厦一切数学成果可建立在集合论基础上！
- 罗素的故事：塞尔维亚有一位理发师：他只给所有不给自己理发的人理发，不给那些给自己理发的人理发。那他要不要给自己理发呢？
- 罗素悖论： S 由一切不是自身元素的集合所组成，那么 S 是否属于 S 呢？

数学危机

第三次数学危机 十九世纪下半叶

- 康托尔集合论 VS. 罗素悖论（理发师悖论）
- 从自然数与康托尔集合论出发可建立起整个数学大厦一切数学成果可建立在集合论基础上！
- 罗素的故事：塞尔维亚有一位理发师：他只给所有不给自己理发的人理发，不给那些给自己理发的人理发。那他要不要给自己理发呢？
- 罗素悖论： S 由一切不是自身元素的集合所组成，那么 S 是否属于 S 呢？
- 1931 年哥德尔不完备性定理：任何一个数学系统，只要它是从有限的公理和基本概念中推导出来的，并且从中能推证出自然数系统，就可以在其中找到一个命题，对于它我们既没有办法证明，又没有办法推翻。 $\Rightarrow$ 把数学彻底形式化的愿望不可能实现

可计算问题

一个系统中有一些问题无法证真证伪而有些可以，那么如何判断一个未解的问题是否有解？

可计算问题

一个系统中有一些问题无法证真证伪而有些可以，那么如何判断一个未解的问题是否有解？

可计算问题

设函数 f 的定义域是 D ，值域是 R ，如果存在一种算法，对 D 中任意给定的 x ，都能计算出 $f(x)$ 的值，则称函数 f 是可计算的。

可计算问题

一个系统中有一些问题无法证真证伪而有些可以，那么如何判断一个未解的问题是否有解？

可计算问题

设函数 f 的定义域是 D ，值域是 R ，如果存在一种算法，对 D 中任意给定的 x ，都能计算出 $f(x)$ 的值，则称函数 f 是可计算的。

研究思路

为计算建立一个数学模型，称为计算模型，然后证明，凡是这个计算模型能够完成的任务，就是可计算的任务。

可计算问题

一个系统中有一些问题无法证真证伪而有些可以，那么如何判断一个未解的问题是否有解？

可计算问题

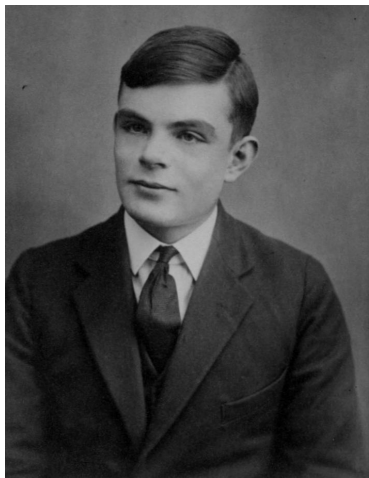
设函数 f 的定义域是 D ，值域是 R ，如果存在一种算法，对 D 中任意给定的 x ，都能计算出 $f(x)$ 的值，则称函数 f 是可计算的。

研究思路

为计算建立一个数学模型，称为计算模型，然后证明，凡是这个计算模型能够完成的任务，就是可计算的任务。

图灵就提出了这样一个计算模型……

图灵与图灵机



图灵机

1936 年，图灵（Alan Mathison Turing, 1912-1954）在其著名论文《论可计算数及其在判定问题上的应用》提出了一种理想的计算机器的数学模型——图灵机（Turing Machine）。

美国计算机协会（ACM）于 1966 年设立图灵奖

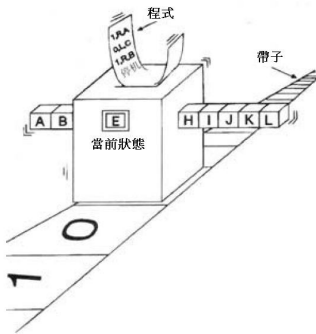
图灵

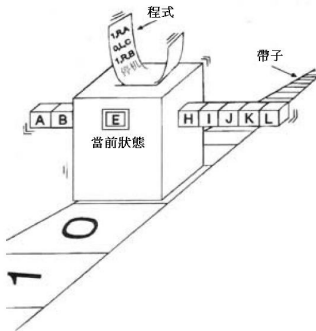
- 1912 年 6 月，生于伦敦。
- 中学期间，获“**国王爱德华六世数学金盾奖章**”。
- 1935 年，被选为“**剑桥大学国王学院院士**”。
- 1936 年 5 月，提出**图灵机**（发表于《伦敦数学会文集》，计算机科学之父）。
- 1938 年，美国普林斯顿大学获博士学位。
- 1938-1945 年二战期间，密码破译工作（任英美密码破译部门总顾问）。
- 1946 年，获“**不列颠帝国勋章**”。
- 1950 年，提出著名的“**图灵测试**”。
- 1950 年 10 月，发表论文《机器会思考吗》（人工智能之父）。
- 1951 年，被选为“**英国皇家学会会员**”（家族第四位）。
- 1952 年，图灵写出一个国际象棋程序。
- 1954 年，逝世（“苹果”）。

内容提要 I

- 1 计算机的基本原理
 - 从数学危机到图灵机
 - 图灵机的构成与原理
 - 计算机为什么能计算
- 2 计算机的历史与未来
 - 历史上的计算设备
 - 从电子管到云计算
- 3 程序运行的基本原理
 - 冯诺伊曼式计算机
 - 存储器的种类特点原理类型
 - CPU 指令的执行
 - 程序的执行
- 4 操作系统与编程语言
 - 操作系统
 - 编程语言

图灵机的组成





图灵机如何工作

① 准备

- ① 存储带上符号初始化
- ② 控制器设置好自身当前状态
- ③ 控制器置于起始位置
- ④ 准备好工作程序

② 反复执行以下工作直到停机

- ① 读写头读出存储带上当前方格中的字母/数字
- ② 根据自身当前状态和所读到的字符找到相应的程序语句
- ③ 根据相应程序语句做三个动作：
 - ① 在当前存储带方格上写入一个相应的字母/数字
 - ② 变更自身状态至新状态
 - ③ 读写头向左或向右移一步

图灵机示例

图灵机示例.key

图灵机停机意味着什么？

- “停机”表示计算完毕：当前存储带上保留的是计算结果。
- 也就是说，“停机”意味着得出计算结果！
- 也就是说，

对于一个问题的输入 A ，问：

A 能否推证出 B ？

思路

如果能找到一个**图灵机**，得出对应的符号序列 B ，那么从 A 到 B 就是可计算的。否则，该问题不可计算。

图灵机示例

图灵机示例.key

图灵机停机意味着什么？

- “停机”表示计算完毕：当前存储带上保留的是计算结果。
- 也就是说，“停机”意味着得出计算结果！
- 也就是说，

对于一个问题的输入 A ，问：

A 能否推证出 B ？

思路

如果能找到一个**图灵机**，得出对应的符号序列 B ，那么从 A 到 B 就是可计算的。否则，该问题不可计算。

图灵机示例.key

- “停机”表示计算完毕：当前存储带上保留的是计算结果。
- 也就是说，“停机”意味着得出计算结果！
- 也就是说，

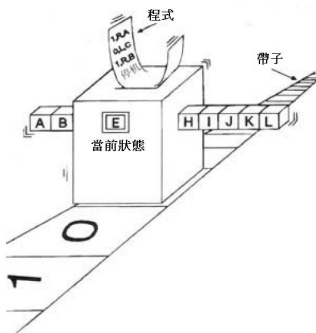
对于一个问题的输入 A , 问:

A 能否推证出 B ?

思路

如果能找到一个**图灵机**，得出对应的符号序列 B ，那么从 A 到 B 就是可计算的。否则，该问题不可计算。





图灵机为什么受到重视？

- 简单！
- 强大！
- 可实现！

图灵机为什么受到重视？

- 简单！
- 强大！
- 可实现！

<http://aturingmachine.com>





A Turing Machine - Overview.mp4

图灵机为什么受到重视？

- 简单！
- 强大！
- 可实现！

<http://aturingmachine.com>

内容提要 I

- 1 计算机的基本原理
 - 从数学危机到图灵机
 - 图灵机的构成与原理
 - 计算机为什么能计算
- 2 计算机的历史与未来
 - 历史上的计算设备
 - 从电子管到云计算
- 3 程序运行的基本原理
 - 冯诺伊曼式计算机
 - 存储器的种类特点原理类型
 - CPU 指令的执行
 - 程序的执行
- 4 操作系统与编程语言
 - 操作系统
 - 编程语言

计算机为什么能计算？

- “数”在计算机中是如何表示的？
- 逻辑上“数”是如何计算的？
- 物理上“数”的计算是如何实现的？

计算机中数的表示

若采用字母表 $\{1, b\}$

符号“ b ” 表示计算对象和计算结果的**边界**

符号“ 1 ” 表示计算对象和计算结果的**数值**

计算机中数的表示

若采用字母表 $\{1, b\}$

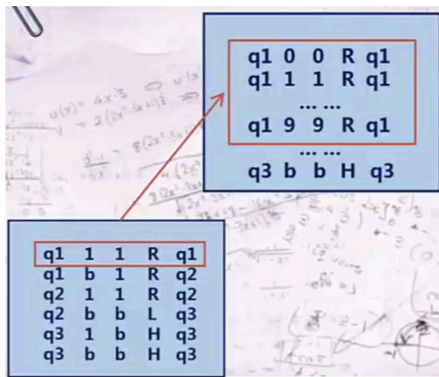
符号“ b ” 表示计算对象和计算结果的**边界**

符号“ 1 ” 表示计算对象和计算结果的**数值**

遇到的问题

- 使用符号“ 1 ”表示数字 1
 - ▶ 数值 1,000,000 应由一百万个 1 来表示
 - ▶ 读入这一个数读写头既要**移动一百万次**
- 显然不合理、不实际

计算机中数的表示



如果使用“十进制”

- 字母表中包含11个符号：
 $\{0, 1, \dots, 9, b\}$
- 用于图灵机控制的程序要大量增加
- 确定当前指令也需要更多的时间

计算机中数的表示

结论

- 字母表中的**符号越多**，读入**移动次数减少**，但**程序数量就越多**
- 字母表中的**符号越少**，**程序量会减少**，但**读入移动次数就越多**

计算机中数的表示

结论

- 字母表中的**符号越多**，读入**移动次数减少**，但**程序数量就越多**
- 字母表中的**符号越少**，**程序量会减少**，但**读入移动次数就越多**

字母表与状态

- 字母表中符号的最优数量，**可能是欧拉常数 e** (2.7182818284590...), 取整后为 3。
- 与具有**两个状态**的电子元件相比，具有**三个状态**的电子元件在制造上更困难，可靠性更低。

十进制、二进制、十六进制

十进制

- 计数符号：0、1、2、3、4、5、6、7、8、9

- 基数：10 $256 = 2 \times 10^2 + 5 \times 10^1 + 6 \times 10^0$

二进制

- 计数符号：0、1

- 基数：2 $10110 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$

十六进制

- 计数符号：0、1、2、3、4、5、6、7、8、9、A、B、C、D、E、F

- 基数：16 $(ABCD)_{16} = A \times 16^3 + B \times 16^2 + C \times 16^1 + D \times 16^0$

十进制转换二进制

除二取余、倒序排列 (正整数)

整数部分:	2	25	余数	小数部分:	0.3125	取整
	2	12	1	$\times$	2	
	2	6	0		0.6250	0
	2	3	0	$\times$	2	
	2	1	1		1.2500	1
		0	1	$\times$	2	
					0.5000	0
				$\times$	2	
					1.0000	1

$$(25.3125)_{10} = (11001.0101)_2$$

二进制转换八进制、十六进制

二进制到八进制

- 从右向左，每三位进行一次转换
- $1111011_{(2)} = 173_{(8)}$

二进制到十六进制

- 从右向左，每四位进行一次转换
- $1111011_{(2)} = 7B_{(16)}$

已解问题：数的表示——二进制

待解问题：如何进行计算？

计算机中数的逻辑运算方法

布尔代数

- 1854 年**布尔**发表《思维规律的研究——逻辑与概率的数学理论基础》和《逻辑的数学分析》创立**布尔代数**
- 为计算机开关电路设计提供了重要的数学方法和理论基础

逻辑运算

- 基本逻辑运算：与、或、非
- 复合逻辑运算：同或/异或、与非、或非、与或非

计算机中数的逻辑运算方法

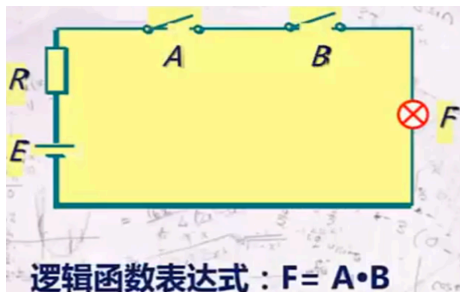
布尔代数

- 1854 年**布尔**发表《思维规律的研究——逻辑与概率的数学理论基础》和《逻辑的数学分析》创立**布尔代数**
- 为计算机开关电路设计提供了重要的数学方法和理论基础

逻辑运算

- 基本逻辑运算：与、或、非
- 复合逻辑运算：同或/异或、与非、或非、与或非

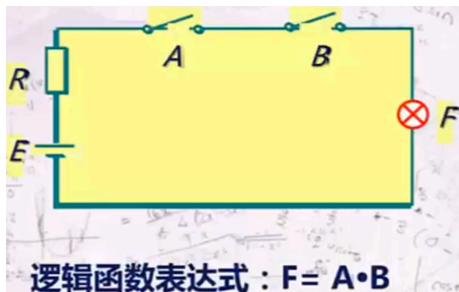
与、或、非



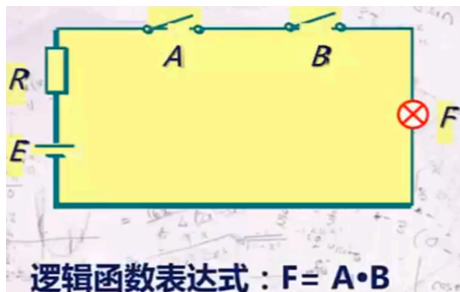
真值表—与

A	B	F
0	0	0
0	1	0
1	0	0
1	1	1

与、或、非



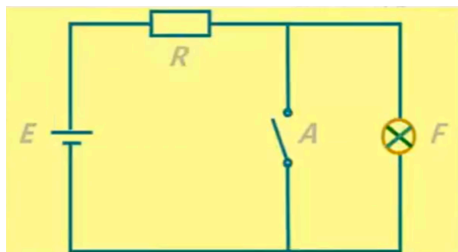
与、或、非



真值表—或

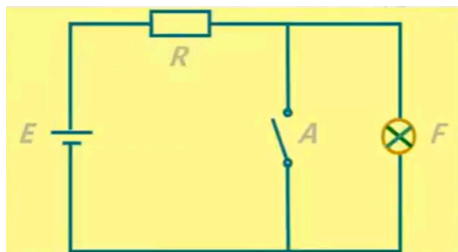
A	B	F
0	0	0
0	1	1
1	0	1
1	1	1

与、或、非



逻辑函数表达式： $F = \bar{A}$

与、或、非



逻辑函数表达式： $F = \bar{A}$

真值表—非

A	F
0	1
1	0

异或/同或

异或

$$F_1 = A \oplus B$$

- 两数相同为“0”
- 两数相异为“1”

同或

$$F_2 = A \odot B$$

- 两数相同为“1”
- 两数相异为“0”

真值表

A	B	F_1	F_2
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

加法的计算

求： $A + B = 1101_{(2)} + 1001_{(2)}$

加法的计算

求： $A + B = 1101_{(2)} + 1001_{(2)}$

$$\begin{array}{r}
 \\
 \\
 + \\
 \hline
 1 1 1
 \end{array}$$

加法的计算

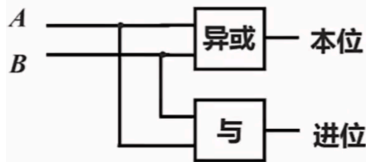
求： $A + B = 1101_{(2)} + 1001_{(2)}$

$$\begin{array}{r}
 \\
 \\
 + \\
 \hline
 1 1 1
 \end{array}$$

$$\begin{array}{r}
 \\
 \\
 + \\
 \hline
 1
 \end{array}
 \quad
 \begin{array}{r}
 \\
 \\
 + \\
 \hline
 0
 \end{array}$$

加法的计算

求： $A + B = 1101_{(2)} + 1001_{(2)}$

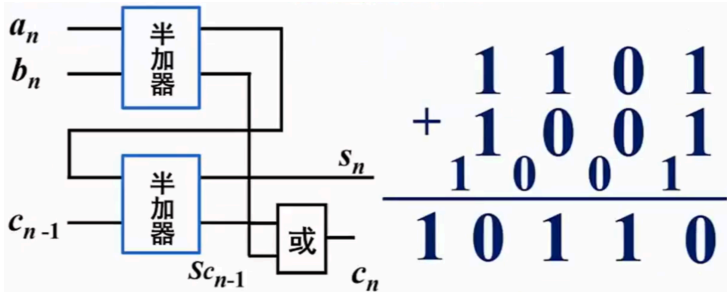


1	1	1	1
+	1	+	0
1		0	
0	1	0	1

0	0	0	0
+	1	+	0
0		0	
1	1	0	0

加法的计算

求： $A + B = 1101_{(2)} + 1001_{(2)}$

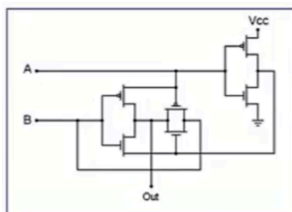


已解决问题：数的表示——二进制

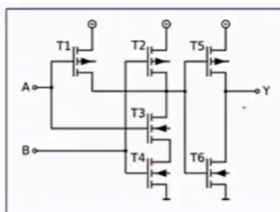
已解决问题：计算方法——布尔代数

待解决问题：布尔运算如何实现？

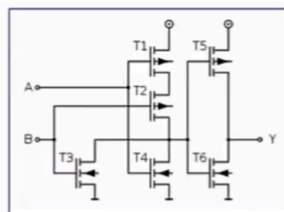
加法的计算



异或门



与门



或门

总结

因为

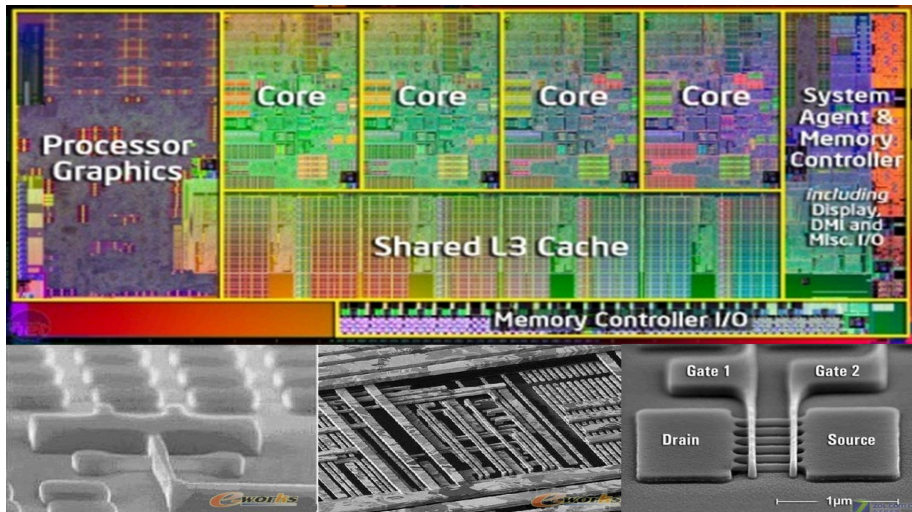
- 参与运算的数可以转换为二进制数
- 二进制数运算可以运用基本的“布尔运算”实现
- 基本的“布尔运算”都可以由电路实现

总结

因为

- 参与运算的数可以转换为二进制数
- 二进制数运算可以运用基本的“布尔运算”实现
- 基本的“布尔运算”都可以由电路实现

所以电路能“算数”！



- ## 2 计算机的历史与未来

- 操作系统
- 编程语言

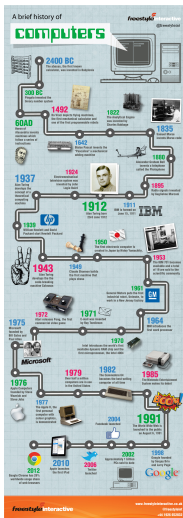
1

2 计算机的历史与未来

- 历史上的计算设备
- 从电子管到云计算

- 操作系统
- 编程语言

先谈历史



早期计算机

- ① 手工计算器：1200 年～1600 年
- ② 机械计算器：1600 年～1930 年
- ③ 计算机原型：1937 年～1946 年

现代计算机

- ④ 电子管计算机：1946 年
- ⑤ 晶体管计算机：20 世纪 50 年代后期
- ⑥ 集成电路计算机：1965 年
- ⑦ 超大规模集成电路：20 世纪 70 年代早期

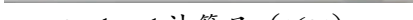


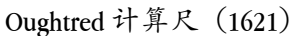
算盘

① 早期的手工计算辅助工具



Napier 乘除器 (1617)





- 无法记录计算法则
- 无法设定计算步骤

- 标记计算过程
- 记录计算结果
- 进行**数字计算**的辅助工具



布莱兹·帕斯卡

- 12岁独自发现了“三角形的内角和等于180度”
- 16岁参加巴黎数学家和物理学家小组
- 17岁时写成《圆锥截线论》震惊数学家笛卡尔
- 18岁时开始设计计算机帮助父亲计算税率税款
- 19岁第三个模型在1642年研制成功

帕斯卡加法器 (1642)

- 一种系列齿轮组成的装置，依靠发条转动，用专用的铁笔拨动转轮以输入数字。
- 开始只能够做6位加法和减法。

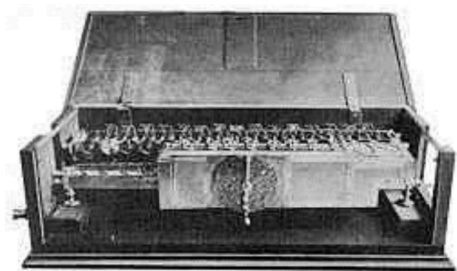
② 1958 年后发现更早的机械计算器



契克卡德 (W. Schickard)

- 德国科学家，时任图宾根（Tubingen）大学教授，广泛涉猎天文学、数学和测量学等诸多领域。
- 1623 年，为自己的挚友、天文学家开普勒（Kepler）制作了一种机械计算机。
- 契克卡德计算机可以进行 6 位数加减法，或许设置了某种“溢出”响铃装置。
- 机器上部附加一套圆柱型“纳皮尔算筹”，因此也能进行乘除运算。

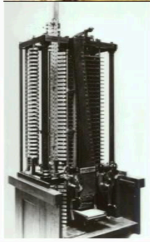
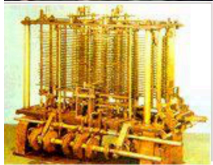
② 工业革命前的机械计算器



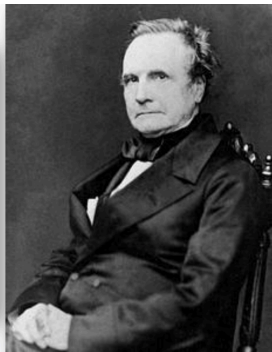
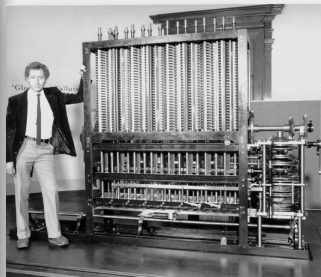
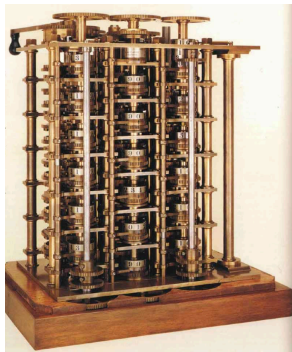
莱布尼茨 (G.W. Leibnitz, 1646
~ 1716)

- 德国伟大的数学家，提出了“二进制”的概念。
- 1673 年在帕斯卡加法器的基础上建造了一台能够进行四则运算的机械计算机，轰动整个欧洲。
- 仍然用齿轮及刻度盘操作，计算结果可以达到 16 位。

② 工业革命时代的机械计算器



- 1822 年英国科学家巴贝奇 (Babbage) 制造出第一台差分机
 - ▶ 可以处理 3 个不同的 5 位数
 - ▶ 计算精度达到 6 位小数
- 1834 年巴贝奇提出了分析机的概念
 - ▶ 机器共分三个部分：堆栈、运算器和控制器。
 - ▶ 企图用机械方式（蒸汽动力）实现计算过程。
 - ▶ 计算用的程序和数据存储于穿孔卡片上。
- 阿达·奥古斯塔 (Ada Augusta)
 - ▶ 为分析机编制了人类历史上第一批计算机程序

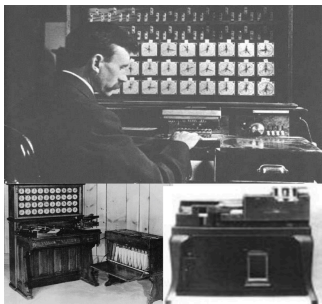


② 机械计算器的不断发展

● 统计学家霍列瑞斯 (Hollerith)

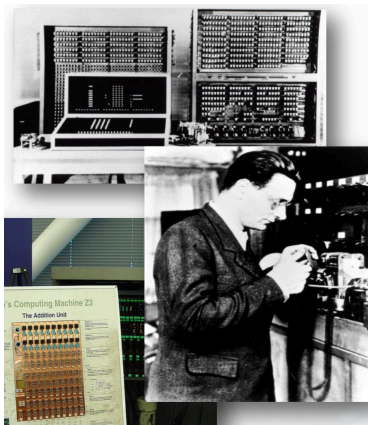
▶ 源自美国人口普查

- 1880 年的人口普查 1887 年才完成
- 1890 年的可能需要到 1900 年才能完成



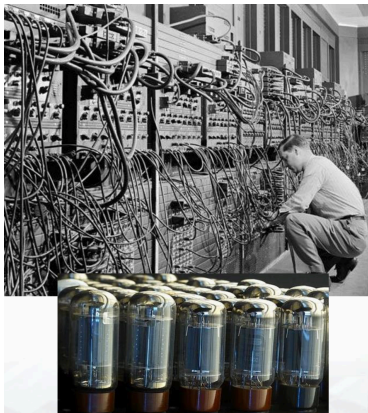
- ▶ Hollerith 制表机 (电子穿孔卡片汇总) 应运而生, 6 个月即完成 1890 年人口普查数据汇总, 随后两年内完成所有统计工作。
- ▶ 1896 年, Hollerith 成立“计算制表记录公司”; 1924 年改名“国际商用机器公司”。

- 1935 年, IBM 制造了 IBM601 穿孔卡片式计算机, 能够在 1s 内计算出乘法运算。



- 1934 年德国工程师楚泽 (K. Zuse) 开始研制 Z1、Z2、Z3 三种型号的计算机
 - ▶ 采用二进制的运算方式
 - ▶ 以继电器为主要元件
- 1941 年 Zuse 完成了 Z3 的研制工作
 - ▶ 第一台**可编程**的电子计算机
 - ▶ 可处理 7 位指数、14 位小数
 - ▶ 使用了大量的真空管
 - ▶ 每秒钟能做 3 到 4 次加法运算
 - ▶ 一次乘法需要 3 到 5 秒

④ 普遍认为的“第一台计算机”



- ENIAC (Electronic Numeric Integrator And Computer)
- 电子数字积分计算机
 - ▶ 宾夕法尼亚大学摩尔学院莫契利 (J. Mauchly) 和埃克特 (J. Eckert)
 - ▶ 1943 年开始、1945 年完成、**1946 年 2 月 14 日**正式启动
 - ▶ 17468 只电子管、7200 个二极管、70000 多电阻器、10000 多只电容器和 6000 只继电器，电路焊接点多达 50 万个，174000 瓦功耗。
 - ▶ 30 米长、3 米高、30 吨重、占地面积 170 平方米。
 - ▶ 每秒 5000 次加法可以在 $3/1000$ 秒时间内做完两个 10 位数乘法。

③ 采用电气元件的“计算机原型”

- 1939 年约翰·阿塔纳索夫 (John Vincent Atanasoff) 制造了 ABC 计算机的第一台样机, 并提出了计算机的三条原则:
 - ① 以二进制的逻辑基础来实现数字运算
 - ② 利用电子技术来实现控制、逻辑运算和算术运算以保证计算速度
 - ③ 采用计算功能和二进制数更新存储功能相分离的结构
- 1942 年约翰·阿纳诺索夫与克里福德·贝瑞 (Clifford E. Berry) 组装出了 ABC 计算机
 - ▶ 采用了二进制数
 - ▶ 将电子管器件作为承载数据的媒体
 - ▶ 设计了逻辑电路
 - ▶ 制造了磁鼓来存储数据并发明了可重复的数据存储方法

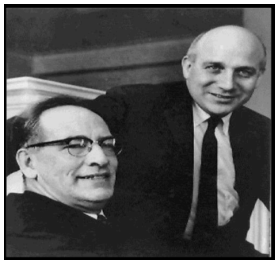


Atanasoff-Berry Computer

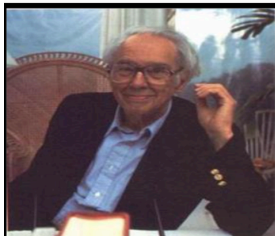


John Vincent Atanasoff

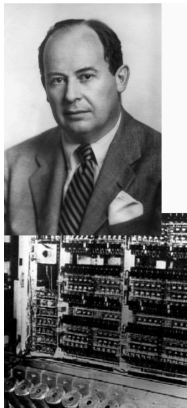
到底是谁发明了计算机？



VS.

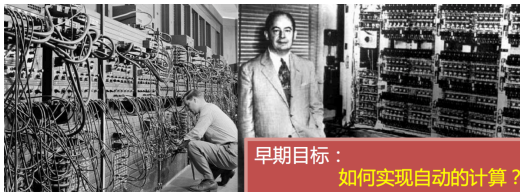


- 第一台计算机的制造者？
 - ▶ 1973 年的 ENIAC 专利之争：美国法院判决 ENIAC 不是 JM Mauchly & JP Eckert 发明的，而是由 ABC 的研究推导出来的。
 - ▶ 背后的故事：Mauchly 拜访了 Atanasoff，并参观了已接近完成的 ABC。
- Atanasoff 由此被称为“电子计算机之父”
 - ▶ 其实也是很有争议
 - ▶ 通常认为“ABC is a near-miss”



- ▶ 1945年3月，计算技术先驱冯·诺伊曼（John von Neumann）来到摩尔学院 Mauchly 和 Eckert 进行了两天讨论，拟定了存储程序式的电子计算机方案。
- ▶ 1945年6月冯·诺伊曼整理发表——存储程序控制原理
- ▶ EDVAC 于 1952 年制造完成
- ▶ 世界上**第一台存储程序计算机**
- ▶ 所有现代计算机的**原型和范本**

计算机发展史



早期目标：
如何实现自动的计算？



当前目标：
计算如何更快速、更方便、更经济

早期计算机

- 手工计算器
- 机械计算器
- 计算机原型

现代计算机

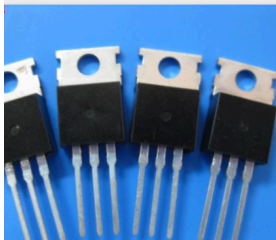
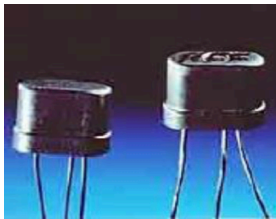
- 电子管计算机
- 晶体管计算机
- 集成电路计算机
- 超大规模集成电路
- 未来的第五代

① 第一代计算机



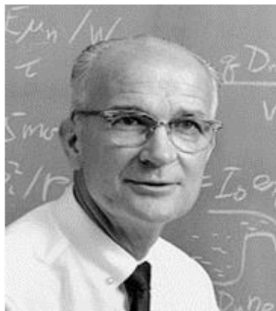
- 时间：20 世纪 40、50 年代
- 主要特点
 - ▶ 使用真空管存储数据
 - ▶ 真空管是一种控制真空中电子流动的电子装置
 - ▶ 被设置为两个状态以表示 0 或 1
 - ▶ 响应比机械快但**体积大**、**耗能高**、**易烧坏**
 - ▶ ENIAC 有 18000 个真空管但第一年运行就替换了 19000 个
- 只能使用 0/1 进行编程

② 第二代计算机



- 时间：始于 20 世纪 50 年代后期
- 主要特点
 - ▶ 使用晶体管存储数据
 - 晶体管由 [贝尔实验室](#) 于 1947 年发明
 - 功能与真空管类似但 **更小**、**更便宜**、**功耗更少**
 - ▶ 产生操作系统
 - 标准化的硬件资源管理
 - 但 **不可移植**
 - ▶ 产生高级编程语言
 - Fortran Cobol

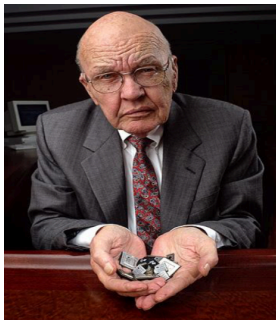
② 第二代计算机



William Shockley
威廉·肖克利
(1910 - 1989)

- 时间：始于 20 世纪 50 年代后期
- 晶体管计算机
 - ▶ 使用晶体管存储数据
 - ▶ 晶体管比电子管更快、更小、更便宜、更可靠
 - ▶ 不需要预热时间、不会产生热量、不会烧坏、不会漏气和爆裂
 - ▶ 电子管需要 1 瓦特功率而晶体管只要百万分之一瓦特
- 1948 年 1 月 23 日 **贝尔实验室**
- 1956 年诺贝尔物理学奖

③ 第三代计算机



Courtesy Texas Instruments

Jack St. Clair Kilby
杰克·基尔比
(1923年 - 2005)

- 时间：始于 1965 年
- 主要特点
 - ▶ 使用集成电路
 - 集成电路可将成千上万的真空管或晶体管压在一个单独的微型芯片上
 - 1958 年由德州仪器公司发明
 - ▶ 操作系统可移植
 - ▶ C 语言产生

③ 第三代计算机

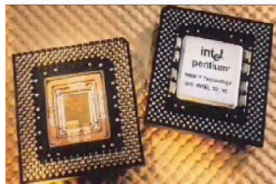


Courtesy Texas Instruments

Jack St. Clair Kilby
杰克·基尔比
(1923年 - 2005)

- 时间：始于 1965 年
 - ▶ 集成电路可将成千上万的真空管或晶体管压在一个单独的微型芯片上
- 1958 年 9 月 12 日德州仪器公司
 - ▶ “我坐在桌子前，待的时间好像比平常晚一点”
 - ▶ “我的工作可能引入了看待电路部件的一种新角度，并开创了一个新领域，自此以后的多数成果与我的工作并无直接联系”
- 2000 年诺贝尔物理学奖

④ 第四代计算机



- 时间：始于 20 世纪 70 年代
- 主要特点
 - ▶ 使用**超大规模**集成电路
 - ▶ **更快、更小、更便宜**
- 第一块微处理器是 1971 年制造的**Intel 4004**
 - ▶ 2400 个晶体管计算能力与 ENIAC 相当
 - ▶ 但尺寸只有 $3\text{mm} \times 2\text{mm}$
 - ▶ 而 ENIAC 尺寸为长 $30\text{m} \times 3\text{m}$ 高
- 计算机发展出现瓶颈
 - ▶ 硬件、软件全方位出现瓶颈
 - ▶ 第五代计算机正在酝酿中……

- 芯片密度每 18 个月增加 1 倍
 - ▶ 1972 年第 1 代 Intel 4004 芯片总共不到 3000 晶体管
 - ▶ Intel Pentium 芯片元件数达到千万数量级
- CPU 性能价格比大约 18 个月翻一番
 - ▶ 芯片工作速度已经达到 10000 MIPS 的工作频率
 - ▶ 速度越来越快而价格越来越便宜

通用电子数字计算机

General-Purpose Electronic Digital Computer

- **通用**：计算机是一种通用信息处理设备，只要有合适的软件，它能适用于各种用途。
- **电子**：是计算机硬件实现的物理基础，计算机的运行最终都通过电子电路中的电流、电位等实现
- **数字化**：是计算机的信息表示方式。一切信息，无论原本是数值、文字、图形、声音等，在计算机里都统一到二进制的数字化表示上。数字化是计算机的一种基本特征，是通用性的重要基础。

计算机分类

微型计算机 (Microcomputer)

- 工作站 (Workstation, 超级微机)：强大的图形功能用于计算机辅助设计
- 个人桌面计算机 (PC)

服务器 (小型机、中型机)

- 速度快、存储容量大、稳定、多个处理器以为多个用户服务
- 可以安装 Windows 系统但大多安装 **UNIX/Linux** 系统
- 速度较快一般用于专业领域的桌面计算机

[HOME](#) [LISTS ▾](#) [STATISTICS ▾](#) [FEATURES ▾](#) [RESOURCES ▾](#) [ABOUT ▾](#)

Search

Go

TOP500 SUPERCOMPUTING SITES



CURRENT LIST
NOVEMBER 2015



TOP #1 SYSTEMS

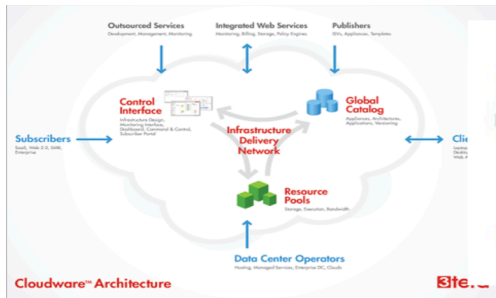
<http://www.top500.org>



<http://www.top500.org>

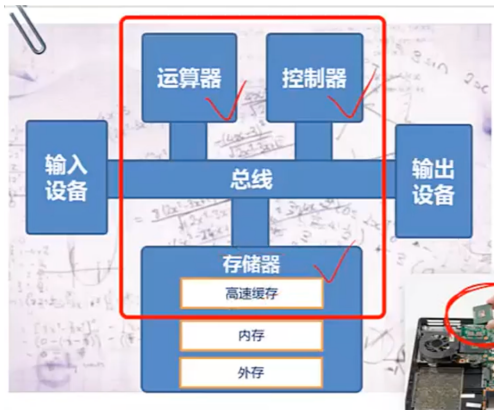


Ranking the World's Most ENERGY-EFFICIENT SUPERCOMPUTERS



- 1 计算机的基本原理
 - 从数学危机到图灵机
 - 图灵机的构成与原理
 - 计算机为什么能计算
- 2 计算机的历史与未来
 - 历史上的计算设备
 - 从电子管到云计算
- 3 程序运行的基本原理
 - 冯诺伊曼式计算机
 - 存储器的种类特点原理类型
 - CPU 指令的执行
 - 程序的执行
- 4 操作系统与编程语言
 - 操作系统
 - 编程语言

- 操作系统
- 编程语言

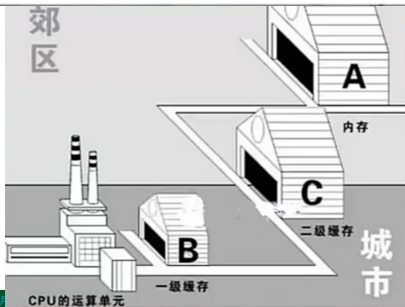
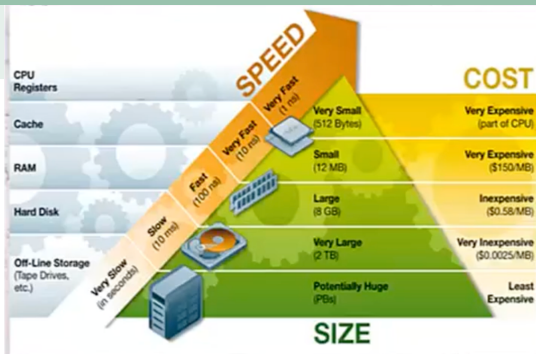


内容提要 I

3 程序运行的基本原理

- 冯诺伊曼式计算机
- 存储器的种类特点原理类型
- CPU 指令的执行
- 程序的执行

- 操作系统
- 编程语言



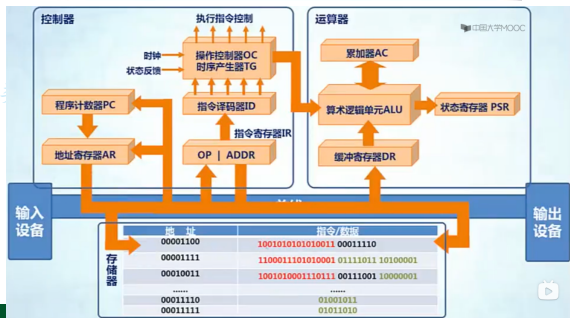
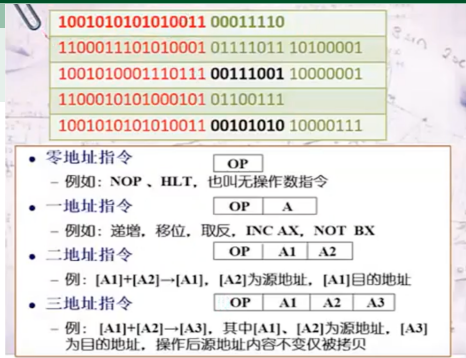
CPU 指令的执行

内容提要 I

3 程序运行的基本原理

- 冯诺伊曼式计算机
- 存储器的种类特点原理
- CPU 指令的执行
- 程序的执行

- 操作系统
- 编程语言



内容提要 I

3 程序运行的基本原理

- 冯诺伊曼式计算机
- 存储器的种类特点原理类型
- CPU 指令的执行
- 程序的执行

- 操作系统
- 编程语言



内容提要 I

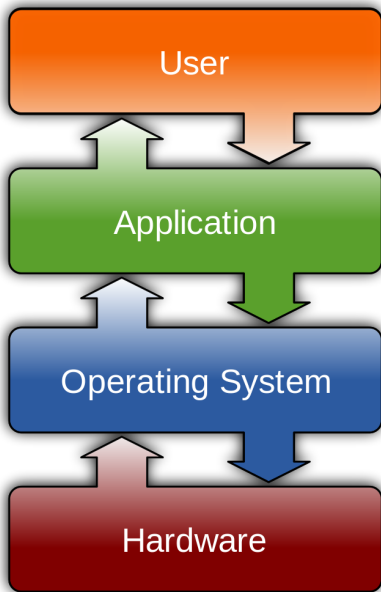
4 操作系统与编程语言

- 操作系统
- 编程语言

内容提要 I

4 操作系统与编程语言

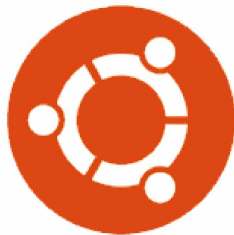
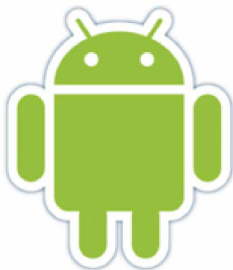
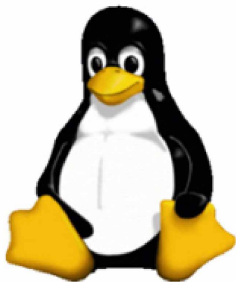
- 操作系统
- 编程语言



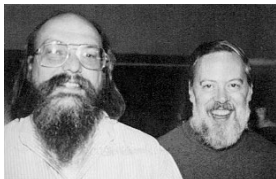
Types

- Single- and multi-tasking
- Single- and multi-user
- Distributed
- Templated
- Embedded
- Real-time
- Library

69 / 86



Unix 起源



MULTICS 1960s

- MULTiplexed Information and Computing Service
- AT&T GE MIT Bell Labs
- General-Purpose Time-Sharing Multi-User on GE-645
- Game “Space Travel” by **Ken Thompson**

Unix

- 1969: “Unics” for “Space Travel” by **Ken Thompson** & **Dennis Ritchie** on Digital PDP-7
- 1973: BCPL (Basic Combined Programming Language) $\Rightarrow$ B $\Rightarrow$ C
- 1974: “The UNIX Time Sharing System”

Unix 分裂

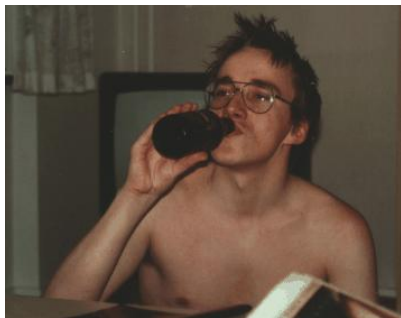
- 1978: BSD (UC Berkeley) VS. USG (AT&T)
- 1979: Unix V7 “the last real Unix”
- 1982: UNIX System V (AT&T) VS. BSD (TCP/IP) $\Rightarrow$ SunOS (Bill Joy)
- 1986: POSIX (Portable Operating Systems Interface)
- 1991: BSD $\Rightarrow$ 386BSD FreeBSD OpenBSD NetBSD DragonFlyBSD . . .

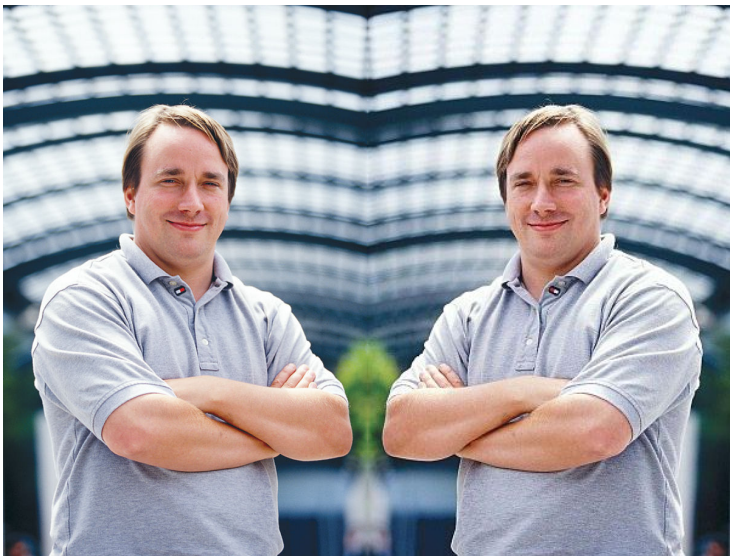
GNU 开源组织

FREE SOFTWARE IS FREEDOM



A photograph showing a man with a long beard and a blue shirt using a laptop. He is standing in a museum or gallery, with other people visible in the background. A large golden bronze vessel is prominent in the background. The date 2005.09.18 is printed in the bottom right corner.





unix_a4.pdf

- Everything (including hardware) is a file
- Configuration data stored in text
- Small, single-purpose program
- Avoid captive user interfaces
- Ability to chain program together to perform complex tasks

Unix 的影响和哲学

KISS = Keep It Simple, Stupid

- Write programs that **do one thing and do it well**
- Write programs to work together
- Write programs to handle text streams, because that is a universal interface.

Unix 痛恨者手册

“The Unix-Haters Handbook”

<http://research.microsoft.com/~daniel/uhh-download.html>

Linux 发行版





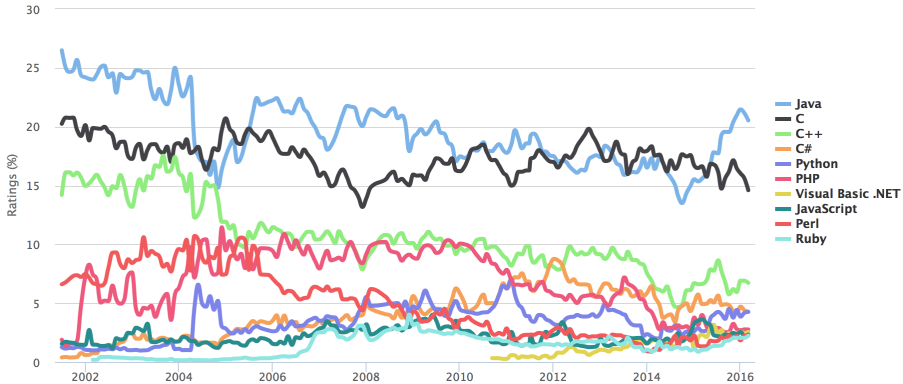
- 组号 $\Rightarrow$ 机号
- 开机选择Linux
- 登录用户名：学号 默认密码：学号（登录后passwd修改密码）
- Linux 常用命令：ls cd mkdir cp mv rm man
- 程序编辑器：vim
- 程序编译工具：gcc g++

内容提要 I

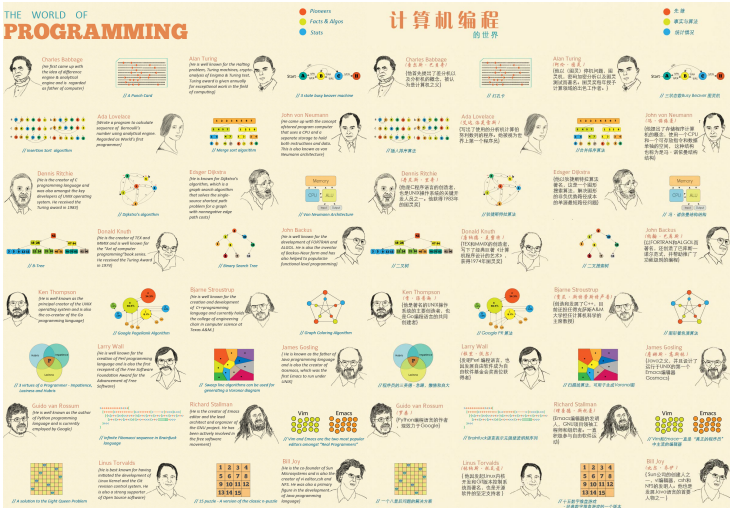
4 操作系统与编程语言

- 操作系统
- 编程语言





编程语言



Timeline



程序员技术练级攻略¹

前言

- 不要乱买书，不要乱追新技术新名词。
- 回顾一下历史，你才能明白明天会是什么样。
- 一定要动手，例子不管多简单，都建议至少自己手敲一遍。
- **一定要学会思考**，思考为什么要这样，而不是那样。
- **还要举一反三的思考！**

¹来源：<http://CoolShell.cn>

- 现在的用户界面主宰：Web 和移动设备 iOS 或 Android
- 低成本高性能 Linux 及各种开源技术架构胜过高成本 Windows
- 微软产品变化太快，很不持久～玩弄程序员？
- 趋势：前端 Web+ 移动，后台 Linux+ 开源。

程序员技术练级攻略¹

启蒙入门

- 1 学习一门脚本语言（如 Python/Ruby）
- 2 用熟一种程序员的编辑器（不是 IDE）和一些基本工具
- 3 熟悉 Unix/Linux Shell 和常见的命令行
- 4 学习 Web 基础（HTML/CSS/JS）+ 服务器端技术（LAMP）

¹来源：<http://CoolShell.cn>

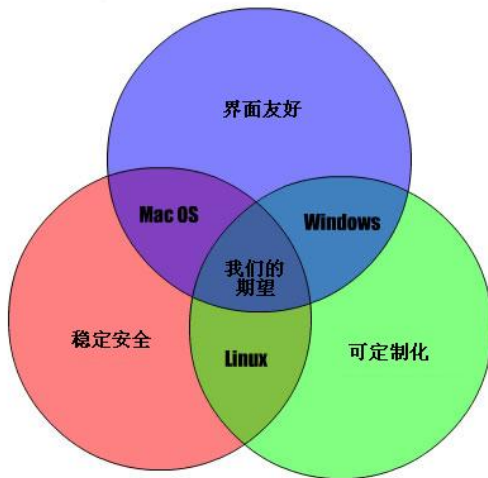
- 1 学习一门脚本语言（如 Python/Ruby）
- 2 用熟一种程序员的编辑器（不是 IDE）和一些基本工具
 - ▶ Vim / Emacs / Notepad++，学会如何配置代码补全、外观、外部命令等。
 - ▶ Source Insight（或 ctag）
 - ▶ Not Only Cool! 更有效率！
- 3 熟悉 Unix/Linux Shell 和常见的命令行
- 4 学习 Web 基础（HTML/CSS/JS）+ 服务器端技术（LAMP）

程序员技术练级攻略¹

进阶加深

- 1 C 语言和操作系统调用（指针、内存模型，数据结构与算法）
- 2 学习 Java
- 3 Web 的安全与架构
- 4 学习关系型数据库（MySQL）
- 5 一些开发工具（Git 或 SVN、gdb、Makefile 等）

¹来源：<http://CoolShell.cn>



粉丝眼中的操作系统



**Mac
粉丝**

**Windows
粉丝**

**Linux
粉丝**

程序员眼中的编程语言

