



以史为鉴，编程语言，启示录之系统觉醒



BtOfficer

什么都不懂的CPA

创作声明：内容包含虚构创作 ∨

5 人赞同了该文章

前言

本篇紧承《明比阅历浅，暗拼体力衰，指下讲武德，码上笑春风》，继续给在途的朋友分享开发途径的格局。开发语言林林总总，要想战无不胜，找到合适的技术路线才是王道。要想利用计算机为生产服务，知己知彼方是上策。没有过时的语言，只有趁手的工具。拿排名的鸡毛当令箭，实践不埋单。本篇站在足够长的时空背景上，审视计算机的开发语言与系统之间关系，希望能给读者带来新的启发。

一、原始混沌



上古法器，尽收眼底

1、人之初，手指头。十进制为什么让人天生喜爱？伸手一数，出脚有份。在玩石头的上古时代，堪称完美。直到现在，数手指头，都是小朋友们的灵光一道。对于成年人，能掐指一算的，都是世外高人。先人们留下的上古法器，谁用谁知道。指针，成了程序世界不可或缺的定海神针，不仅是十进制机械时代的肉身，更是二进制电子化的灵魂。没有指针的语言，不是好语言。

2、指头不够，石头来凑。进制毕竟是被虐千千万后的产物，上古人还没摸到一手数，一手记的门道，甚至不看脚上有。进化使然，他们打量起了手上的石头。万万没想到，小石头竟让人脱离了自身的束缚，成了计算机的鼻祖。数石子儿，管够！直到今天，人类对石头仍然执迷，人难忘本啊。

3、石头的进化，由硬到软。趁手石头难再有，笨重易丢，可移植性太差。思来想去，左顾右盼，还是裤腰上的麻绳合适。挽疙瘩，可复写，讲究多，语义丰富。挂脖子，悬房梁，轻便。送酋长，赠姑娘，简直不要太好。谁又敢说链表结构，没有借鉴麻绳？如果一直发展下去，麻绳或将是首个兼顾盲人的IDE，兼用眼神和触感神经输入，人脑解释编译并运行的高级计算机系统。可惜了，人类再也回不去麻绳时代。

4、曲折的尽头，规则还是硬的好。麻绳是好，遇火难解。各有说道，道难再道。很明显，松软的规则，容易挖墙角。于是，算筹来了，咱不数指头，数棒棒。如果不是石头和麻绳，在前面挡了又挡，让人痛定思痛，今天给大家分享的可能就是”棒棒的一万重天”。

棒棒对人类的影响颇为深远。粗可盖房梁，细可煮生汁。长可驱走兽，短可数棒棒。谁又敢说，条形码不是数棒棒的遗物。最重要的是，人们在码弄的时候，发现了进制。

1500年前，东边的算盘，迈着轻盈的步伐，朝着账本走来。精巧的外壳，多档位存储器、规整的地址总线，灵活的指针（手指），健壮的计算（口诀），在人脑的驱动下，算珠跳着欢快的舞蹈，一直活到今天。她时刻告诫世人，简便、易用、健壮的，往往命长。

18世纪末，西边的瓦特在蒸汽机的捶打下，成功锻造出计算尺。自此，计算机终于拥有了钢铁之躯，硬核时代就此展开。事实也证明，硬核需要千锤百炼，没有硬核的系统，不是好系统。摸不到内核的程序员，不是好码手。

二、机械学步

当东边拨弄着算珠，沉醉于没有什么需要去掉的时候，西边蒸汽腾腾，如火如荼的机械发明前仆后继。1674年，法国数学家莱布尼茨，在帕斯卡加法器的基础上，研制了可加减乘除的机械式计算机，并率先提出二进制。

在这之前，计算机的作用，仅是为了展开算法执行的中间过程，以便观察（调试）。谈不上人机交互，人的思想(算法)也无法投射。直到1725年，为解决纺织业中图案机织问题，法国机械师布乔发

明了“穿孔纸带”，1805年被法国机械师**杰卡德**改造为穿孔卡片，才开创了**程序设计的**先河。优雅的爱美，灵感的源头。正所谓“踏破铁鞋无觅处，得来全不费工夫，无心插柳柳成荫”。

1822年，受杰卡德穿孔卡片的灵感，法国**巴贝奇**的差分机诞生了。此后在拜伦独生女阿达的资助下，二人展开了对分析机的研发，阿达也成为公认的首位软件工程师，其编写的函数指令被后来的ADA语言所吸收。虽然分析机最终失败，但其结构理论在后续100多年里，都指导着计算机的改进。历史证明，光有情怀是不够的，摆出结果才令人信服。

1890年，德国侨民**霍列瑞斯**，在美国人口普查时，借助杰卡德和巴贝奇的发明，开创了计算机**数据处理**的先河。后来，他创办的制表机公司成为IBM的前身。道高一尺，魔高一丈，代码的千面万像里，应用才是王道。

三、电气之舞

1906年，德福雷斯特发明了真空三极管（电子管），为计算机步入电气化打下基础。

1942年，阿塔纳索夫和克利福德·贝瑞设计的ABC计算机，确定了二进制算术和电子开关，为第一台通用电子计算机的诞生奠定了基础。

1944年，艾肯在IBM的资助下，站在前人研究的基础上，借助电流和继电器进行计算的MARK II问世，这是世界上第一台电子计算机，其程序员就是大名鼎鼎的霍波女士。因虫子阻断电流触点而导致故障，她将程序故障称之为**Bug**，流传至今。Bug也揭示出指针虽好，安全第一。

1946年，在美国军方资助下，莫希利和埃克特研究组研制的第二台电子计算机ENIAC成功问世。一改MARK I的机电模式，在借鉴了ABC机器的电子构建思路的情况下实现了全电气化，计算性能提升了上千倍。尽管如此，采用电子管的ENIAC仍然非常笨重，为了改进，冯·诺依曼等人提出了著名的101报告，确定了以二进制为基础的现代计算机的基本物理结构。冯·诺依曼跻身计算机之父，说明凡事多总结，站在巨人的肩膀上，不当轮子哥。

1947年，美国贝尔实验室的肖克利、巴丁和布拉顿，发明了晶体管并首次实用，使得计算机摆脱了机械，完成了电子化的蜕变。

1950年，英国的图灵发表了《计算机与智能》，确定了后续计算机的思想原则，开创了人工智能。他还提出了**存储程序**的概念，促进了操作系统的诞生。要不是天妒英才，科幻中的很多场景可能早已实现。

1951年，英国的威尔克斯，发展了**微程序**概念，大大简化了CPU的研发。他提出的**符号标志，宏和子程序库**等概念，为后面操作系统和高级程序语言的发展提供了路径。

四、系统觉醒

1、创业早期，既是领导又是兵。

在机械化、机电化时代，计算机身形笨重而巨大，为科学研究特定问题而生。即便如此，成品也非常稀少，手工输入定制/固化的指令就足够了。为了解决手工低速导致的效率低下和机器等待，预置指令的脱机系统，开始在机电化的计算机上得以应用。

这一时期，开发语言是直面机器的，机器支持什么样的指令，就用什么样的指令。不但需要理解机器的原理，还需要维修排错，也只有机器的制造者能够胜任。随着机器制造工艺的稳定，后面才逐步发展出了指令助记语言-汇编语言。

这也说明，指令是机器的化身，是计算机上真正唱歌的角儿。

2、资源不足，每一分都要在刀刃上。

在晶体管时代，计算机的体积和功耗得到大幅缩减、性能却显著提升，为进一步实用夯实了基础。但制造成本仍然极其昂贵，出产极少，主要用于科学研究中的数据处理。为了充分利用这些昂贵的设备，这就需要计算机存储程序指令，用以压缩每道作业程序的空闲时间，这样基于空闲监视的作业调度、资源分配的单/多道批处理程序开始出现。

尽管如此，批处理的需求却十分简单，还谈不上是个事儿。计算机的使命，仍然是如何进行高效的计算，计算机的大部分使用者都是科学家，编程需要直奔主题。汇编虽然高效，但仍然是直面机器的。毕竟科学家不是程序员，专注科研生产才是正道。围绕解决科学家们使用汇编语言的难题，面向计算函数的（面向过程）语言，如Fortran、Cobol、Algol、C等，应运而生。

由此可见，语言服务于机器，更服务于实际问题。

3、生意红火，职能分工，战略与实施才能两不误

1956年，硅台面晶体管问世，开启了集成电路的大门。此后，按照摩尔定律，计算机的体积和能耗越来越小，性能越来越高。机电时期的卫星机功能，随着工艺的进步，被逐步缩小集成到了核心机上，为计算机的普及应用奠定了硬件基础。

计算机的功能越来越完善，开始介入科学计算以外的领域，也让其应用越来越复杂。以前的人工方式被彻底剥离，转而由各种接口替代，人机交互的矛盾愈演愈烈。但是科学家们只想写计算函数搞研究，不想成为程序员。如何整合简化这些硬件资源的使用，又不影响科学家们写计算函数，就成了亟待解决的问题。

简单的批处理，不仅无法交互，其伪并行也无法从根本上解决资源的浪费，更无法解决硬件升级后复杂的协调工作。一个具有多道批处理、分时或实时特征，专注于为应用开发服务的通用程序，呼之欲出。

自此，系统开始觉醒，系统编程和应用编程开始分工分离，不仅催生出更广泛的程序员职业，还让开发语言从面向过程进化到面向对象。而这一切的根本诱因，正是硬件日益复杂和完善下的人机交互。可见人机交互的便捷性，才是人道主义。

五、启示录

编程语言早于系统，系统是编程语言的产品，只不过它常驻内存，为应用服务而已。从机器码到助记汇编，从面向过程到面向对象，只不过追随硬件罢了。助记汇编机器的化身，面向过程心向计算，面向对象才是丛林。各自有各自的历史使命，后来者只不过肩负起了前者，负重前行。代表机器的汇编不会死，专心于计算任务的面向过程也不会死，反倒是负重的面向对象，在激烈的竞争中就不准了。

编辑于 2022-01-23 12:11

启示录

编程语言

编程

写下你的评论...



还没有评论，发表第一个评论吧

文章被以下专栏收录



BtOfficer之VB/VBA

VB/VBA荒的很啊，也没啥深度内容，那就来开荒吧

推荐阅读



和小浩学管法

所有的编程语言知识，都包含在这100张思维导图里了 | ...

量子位

发表于量子位

“语言只是工具，编程思维才是最重要的”，我信你个鬼！

【小宅按】还记得我刚学编程那会儿，时不时会在网上看到或者亲耳听到一句话：“语言只是工具，编程思维才是最重要的”。我对此深信不疑。我现在想说：“你这个糟老头子坏得很，我信你个鬼...”

华为云开发者联盟

读《编程语言的设计与实现

此书作者是Ruby语言的创始人：松本行宏一本新书，之前也拜读过他的《松本行宏的程序世界》。看这本书最大的感受就像一个平易近人的老朋友在给你讲故事，把这门编程语言和思考过程娓娓道来...

赵俊民

发表于模式