

计算的历史：世界图景的数字化

A History of Modern Computing 读书笔记

罗云千, 2019012343, luoyq19@mails.tsinghua.edu.cn

在本学期的第十三课上，老师讲述了人们为了进行计算（computing）所发明出的种种机械装置，以及这些装置是如何最终演变成 ENIAC, EDVAC 这样“图灵完备的电子计算机”。而在第十四课上，讲述了互联网是如何从美国军方的研究项目演变成我们所熟悉的形态。但是在这两段历史之间，有一个显著的缺口（gap）：在前一段历史的结尾，EDVAC 还是一台造价近百万美元，专供军方进行弹道计算的巨型机器，当时一位数学家和计算机工程师 Howard Aiken 甚至宣称只需要五六台计算机就能满足整个美国的需求了 [1, p. 13]；而在后一段历史中，计算机变得越来越小，性能越来越强，它不再专属于大型机构，而是成为了个人的所有物；更重要地，它似乎不再是用来“计算”的机器了。计算机是如何偏离它的“本意”的呢？

在 *A History of Modern Computing* 一书的前言中，作者提及了他在 80 年代初期，作为一个计算机历史的研究者遇到同行的一个提问：“为什么是计算机？为什么不去研究洗衣机的历史呢？(Why computing? Why not study the history of washing machines?)”[1, p. 2]。这个提问似乎是平凡（trivial）的，但从另外一个角度来又呈现出不平凡之处：计算机和洗衣机这两项技术的本质区别在哪里？既然像名字所提示的一样，这两个发明一个让计算更快捷，一个让洗衣服更快捷，那么为什么我们比起后者更关注前者呢？

A History of Modern Computing 书中结论（Conclusion）一章的标题——“世界图景的数字化”（The Digitalization of World Picture）向我们提示了这个问题的答案。计算机的发展不只是 CPU 的主频越来越高，内存的容量越来越大。更加关键的问题是，计算机是如何朝着这样一个目标，即“将世界上的一切都纳入数字的领域之中”前进的。而在这样一个过程中，计算机始终作为数字化的媒介，它所起到的作用自然也就和纯粹的计算发生了偏差。而正是由于它的媒介作用，计算机被赋予了不可替代的意义。

在这篇读书笔记中，我们将考察这样一个过程是如何发生的。我们主要分三个阶段来考虑：在第一个阶段，电子计算机刚刚出现，但是它已经给人们带来了完全的“可编程性”，而这种可编程性塑造了计算机数字化和抽象化的基本思想；在第二个阶段，大型机开始向小型机演变，在这个过程中人与机器的交互过程被重新定义了；在第三个阶段，个人电脑开始普及，互联网开始出现，数字化得以侵入社会的运转和每个人的生活。尽管书中更多的是关于历史过程的翔实记叙，但本文希望通过对记叙过程的整理和重构，捕捉一些关键而隐藏的思维变化中的动机和结果。

1 程序的数字化——从硬件到软件的抽象

1.1 von Neumann 架构

Eckert 和 Mauchly 在 Moore 电子工程学校于 1945 年制造完成的 ENIAC 常常被认为是现代计算机的起始 [1, p. 15]，这不仅是因为它的计算速度超过了当时的其它计算机，更因为它具有某种“通用”的特征：人们可以在它上面“编写”程序，在运行程序的时候，ENIAC 从打孔卡片（punched card）中读取数据，将数据存储在电路单元（或者用现代话语来说，内存）中，然后根据预先编好的程序来处理这些数据，通过打孔器输出结果。在 ENIAC 上所编写的程序能够实现丰富的功能。用后来 Turing 提出的术语来说，ENIAC 是一个 Turing 完备的机器。而人们一般认为，不存在一个机器，它能够完成的计算任务比 Turing 完备的机器更多（不考虑计算所需要的时间和存储空间大小的话）。

尽管 ENIAC 是可编程的，但是它的编程方式非常原始，甚至在现在人们眼中算不上是编程：人们需要手动连接迷宫般的线缆，设置大量的开关；每次重新编写程序来解决新问题的复杂程度不亚于配置一台新机器，这样的复杂性大大限制了人们能够在其上所编写的程序 [1, p. 21]。

更加符合现在直觉的编程可能要追溯到 1944 年的 Harvard Mark I，它通过打孔的纸带来编程：编程者在纸带上通过特定的方式打孔来表示将要运行的程序，Harvard Mark I 就像读取数据一样读取打孔纸带上的指令并执行。由于在只需要在纸带打孔而不需要操纵复杂的电路，因此它要比 ENIAC 方便许多 [2]。但是这种将指令存储在纸带上的架构限制了它的功能：它难以实现类似于“跳转”¹的操作，因此可编程的空间比较有限。和 Harvard Mark I 类似的机器还有当时 IBM 的 CPC (Card Programmed Computer，用打孔卡片来进行编程) 系列产品。

Eckert 和 Mauchly 在制造 ENIAC 的过程后期，逐渐产生了一种对计算机架构的新设想：在计算机运行的时候，可以将计算机运行的程序同输入的数据一样存储在电路单元中。通过这样的方式，就可以像 ENIAC 一样运行功能丰富的程序，同时像 CPC 一样方便地编写程序 [1, p. 21]。按照这个架构，Eckert 和 Mauchly 设计了新的计算机 EDVAC，涉猎广泛的科学家 von Neumann 作为这一项目的顾问，对其架构进行了抽象和总结，他在 1945 年做的关于 EDVAC 的报告 First Draft of a Report on the EDVAC 使其广为人知，von Neumann 架构也因此得名。

von Neumann 架构奠定了现代计算机的基础——直至今日，这一架构仍然用于几乎所有的计算机。这个架构有一个显著的不平凡之处：“数据”可以是一些数字、字符，而“程序”则代表一些更抽象的过程或者模式，这两者似乎注定不是同一类的东西；但是对于 von Neumann 架构而言，这两者被使用类似的方式被处理：以同样的方式承载、读取、存储，它们都成为了纸带、卡片、电路中的数字²。这是一种全新的编码，或者说数字化的思想，它不仅像电报那样将数据（即数字、文本等）数字化了，也将过程、模式、操作这样更为微妙的东西数字化了。虽然这种想法在 Harvard Mark I 和 CPC 中已经初现端倪了，但是 EDVAC 所展示出的能力让人们真正看到了这种数字化的强大之处。

1.2 软件与抽象

von Neumann 架构所带来的直接结果就是所谓“软件”的出现。既然程序不再和机器中的电路这样实体化的东西直接相关，而它在表达能力又具有通用性，那么人们自然会试图将程序和机器解耦。一部分人开发程序，另一部分人开发机器，开发程序的人不再依赖于某一台具体的机器，他所编写的程序可以和其它人分享，可以以相对而言很低的成本复制，从而给用户创造了广阔的空间。

50 年代，计算机的商用已经逐渐成熟起来了。IBM 在 1954 年推出的 704 计算机吸引了一批用户组成名为 SHARE 的小组，他们交流经验、分享自己写的子程序 (subroutine)，很快便积累了许多由子程序组成的库 (library)。SHARE 不仅帮 IBM 增长了知名度和销量，其贡献和影响力也使得 IBM 不得不认真考虑其对未来产品提出的建议 [1, p. 88]。

在 SHARE 中诞生了汇编语言 (assembly) 和汇编器 (assembler)。汇编使得人们可以用更加直观的方式编写程序³，从而降低了编写和修改程序的难度。但是由于汇编一般只能适用于同一种架构的机器，人们还希望编写一次程序可以在多个不同架构的机器上运行，于是便诞生了高级程序语言和编译器——高级程序语言提供了一种独立于机器的方式来描述程序的行为，然后编译器将其翻译成能够在多种不同机器上执行的机器码⁴。在 50 年代后期，出现了大量的高级程序语言，FORTRAN 和 COBOL 是其中最具影响力的两个，前者广泛应用于科学计算领域，后者则主要面向商业和政府部门 [1, pp. 90-95]。

从编译器中我们已经能粗略看到数字化所带来的一个本质上的改变：抽象 (abstraction)。抽象意味着将某个东西的内部构造和信息隐藏起来，只对外暴露它的“本质”特征和功能。汇编语言隐藏起不同机器不同的指令和机器码，而只对外暴露了一些用文本描述的指令；编译器隐藏了不同机器架构的区别、内部的原

¹指令一般是按照顺序依次执行的，而跳转 (jump) 指的是从执行当前的指令变为执行不相邻的另外一条指令的操作。而由于卡片是按顺序读取的，因此要想跳转到另外的指令必须调整卡片的位置并重新读取，这个过程会大大降低运行的速度。

²和现在作为事实标准的二进制不同，早期的计算机有许多是十进制的。

³例如汇编代码 `mov eax, ecx` 表示将一个名字为 `ecx` 的寄存器上储存的数据移动到名字为 `eax` 的寄存器上，其对应的机器码可能是 `89C8`。

⁴最早期的 UNIVAC 编译器的作用与此稍有不同，它负责将子程序组合成一个完整的程序 [1, p. 85]。而现代编译器则的作用则往往更为广泛，我们在这里忽略这些差别，只考虑其最普遍的用法。

理，而给程序员只提供诸如“将某个变量乘以另外一个变量”之类的抽象操作方式。这样的思想也许可以追溯到工业时代所产生的标准化零件，人们从各种各样的零件中抽象出某种通用的零件并为其制定标准，供应商只需要提供符合这一标准的零件，而用户也只需要寻找什么标准的零件符合其需求。但是零件还是要依托于实体而存在，而实体会受到各种各样的物理限制，因此这种抽象的级别是非常受限的。但是对于数字化的存在来说，没有了物理上的限制，只需要在“数字”上符合各自的标准即可，因此抽象是可以无处不在、自由叠加的。人们可以先从机器码中抽象出汇编语言，然后从汇编语言中抽象出程序语言，而程序语言中又可以抽象出子程序、库。这种多层级的抽象在计算机中无处不在，正如 David J. Wheeler [3] 指出的：

我们可以通过引入一个额外的中间层来解决任何问题。

We can solve any problem by introducing an extra level of indirection.

从结果来看，抽象最重要的意义就是降低复杂系统的心智负担，在每一层抽象中我们都可以隐藏这一层之下的复杂细节，从而将整个系统的复杂度压缩。因此即使一个系统的复杂程度远远超出人类大脑能够容纳的范围（例如上百万行代码的软件项目，或者以纳米为单位的集成电路），人们依然可以通过抽象的方式进行把握。在之后的发展过程，包括操作系统、互联网中，抽象将继续扮演着核心的地位。

2 交互的数字化——从批处理到分时系统

2.1 计算机工艺的进步

为了避免与学校的知识产权纠纷，Eckert 和 Mauchly 在 1946 年离开学校并创立了 Eckert-Mauchly Computer Corporation (EMCC)，并在 50 年代初开始推出作为 EDVAC 后续的 UNIVAC 系列产品，Mauchly 凭借其敏锐的商业头脑找到了许多客户，包括美国空军、海军、能源公司、保险公司、通用电气等 [1, pp. 24-30]。随着 UNIVAC 的成功，许多其它公司也加入了计算机市场的竞争。

计算机的主要组成部分，包括逻辑电路的种类、架构设计、存储介质等，它们的工艺在 50 到 60 年代之间都发生了巨大的变化。

ENIAC 到 UNIVAC 的一系列产品都使用延迟线存储器 (delay line memory) 作为计算机的内存，而 IBM 以及同时期的 CRC、ERA 等公司的早期产品则主要使用更慢但是容量更大、更可靠的磁鼓存储器 (drum memory)。在 1954 年之后，更快、更可靠、体积更小的磁芯存储器 (core memory) 逐渐成为主流。

早期的电脑基本上都使用真空管来组成其逻辑电路，但是真空管耗电高、容易出故障。隶属于 AT&T 的 Bell 实验室在 50 年代初期开始研究晶体管作为其替代品，由于 AT&T 在通信行业的垄断地位，Bell 实验室不得不将晶体管的技术公开以避免反垄断调查。许多半导体公司因此开始制造晶体管电路，50 年代后期，晶体管的技术逐渐成熟，由于其更小的体积和更低的耗电等因素很快取代了真空管。晶体管不仅用于逻辑电路，它也取代了磁芯存储器作为内存的介质 [1, pp. 64-70]。

德州仪器的 Jack Kilby 和仙童 (Fairchild) 半导体公司的 Robert Noyce 在 1959 年陆续发明了集成电路 (integrated circuit)，集成电路的体积远远小于非集成的电路 (discrete circuit)，因此受到了制造 Minuteman 弹道导弹的美国军方和进行 Apollo 登月计划的 NASA 的青睐。1966 年之后，陆续有使用集成电路的商用计算机出现 [1, pp. 182-190]。

而对于外部存储设备来说，其种类一直多种多样，纸带、打孔卡片、磁带、磁碟都在市场中有一席之地，其中磁带由于其较大的存储密度成为了大型机时代的主流。但是由于磁带只能按顺序读写，在 60 年代后期到 70 年代初，随着 IBM 大幅改进了能够随机读写的磁盘 (disk storage，包括软盘、硬盘等) 的性能和成本，磁盘也逐渐占据了市场 [1, p. 200]

2.2 计算机形态的变革

在计算机工艺升级换代的过程中，其硬件的体积变得更小、成本更低。而这则导致了计算机的形态发生了巨大的变化。早期的计算机的往往面向军方、大型企业、研究所这些需要机械式地处理大量数据而对预算

不敏感的客户，因而这些计算机都价格昂贵（从几十万到上百万美元不等），体积庞大（占地几十到上百平方米），内存大小充足，I/O 装置齐全（键盘、纸带、打孔卡片、磁带）[1, p. 63]。这种形态的计算机到 60 年代逐渐成熟和标准化，被称为大型机（mainframe）。

IBM 在这个时代已经成为了大型机的最大制造商。它在 1964 年发布的 System/360 系列，正如其名字所暗示的一样，试图从低端到高端、从科学计算到商业计算，囊括所有的市场空间（full cycle of computing）[1, pp. 153-154]。通过对架构的精心设计，System/360 系列保持极好的兼容性：为之前的 IBM 产品所编写的代码，不需要进行修改就可以直接在 System/360 上运行。同时，架构上的诸多重大改进也使得其性能和功能大大增强。这一系列取得了极大的成功，IBM 在 60 年代中期股价飙升，市场份额一度达到 70%[1, pp. 144-160]。

尽管 IBM 试图囊括所有市场空间，但是它忽略了一个新兴的市场空间：小型机（minicomputer）。小型机起源于 1960 年 CDC 公司为其推出的 1604 大型机进行辅助处理 I/O 的 160 计算机，而在 DEC 公司手中发扬光大。小型机有诸多特征：在架构上它使用了较短的字长（word length），从而降低了电路的复杂程度和成本；得益于硬件工艺的进步和采用紧凑的封装方式，它的体积也减小到了一台桌子左右的大小；同时，其成本也大幅下降，售价往往低于大型机的十分之一 [1, pp. 124-135]。DEC 在 1965 年发布的小型机 PDP-8 引发了市场轰动：通过使用 Philco 公司微合金扩散（micro-alloy diffused）半导体工艺和对磁芯内存的改进，以及对于架构的巧妙设计，PDP-8 的计算速度甚至不逊色于 IBM 在 50 年代末的大型机 7090（尽管其内存比较有限），而其价格仅需 18000 美元，从而吸引了许多预算不够充足的企业和实验室中的科学家和工程师们。

2.3 计算机交互的变革

对于 60 世纪中期之前的绝大部分计算机来说，“批处理”（batch processing）是使用它的唯一方式：用户将自己的程序和数据提交给计算机，计算机按照提交顺序依次处理不同人提交的程序和数据，一段时间之后用户在打印机处取回程序运行的结果。一个典型的场景可能是这样的：大型机被安置在一个专门的大房间里，房间里的温度和湿度都需要仔细调节来防止机器故障，用户将一沓打孔卡片递给操作员，操作员则将其放到计算机的读取处；在等待其它人的程序几天后运行完之后，用户发现自己只收到了一大摞冗长乏味的核心转储（core dump）报错信息，必须在这些似乎毫无规律的数字中寻找自己程序的错误 [1, pp. 99-101]。

在这个过程中，人和计算机没有任何交互（interactive）[1, p. 99]。批处理过程使得编写和调试程序都十分麻烦。对于商业用户来说，由于计算机需要处理的问题比较固定，这个缺陷并不会造成太多困难。但是对于大学中的科研人员和学生来说，这个缺陷就十分让人困扰了。

60 年代中期，开始出现基于电传打字机（teleprinter）的终端（terminal）用在计算机上：用户通过键盘来输入指令和数据并且通过打字机显示在纸上，当计算机运行时，其输出的结果也通过打字机打印在纸上 [4]。在 60 年代末期，CRT（cathode ray tube，阴极射线管）显示器成本的下降，使得视频终端凭借其更快的输出速度和更丰富的功能逐渐取代了打字机终端 [1, p. 226]。虽然终端改进了批处理操作的体验，但是由于程序依然是按照顺序执行的，交互的实质并没有改变。事实上，直到 60 年代中期，高昂的成本使得任何任何计算机交互而让计算机闲置的时间都过于昂贵了，只有对于不计成本的项目，例如美国空军的 SAGE 防空系统，美国税务局的税务统计系统，或者 NASA 的载人航天计划，交互操作才可能被列入需求清单 [1, pp. 53, 115-124]。例如，在 60 年代初 NASA 的载人航天项目中，一个被称为水银监视器（Mercury Monitor）的装置监控系统的运行情况，并通过输入特定的指令来终止处理器的计算，以防止危险的情况发生。

而改变这一点的，一方面是工艺改进以及小型机出现所造成的价格下降，另一方面是分时（time-sharing）系统在 60 年代中后期的发展。分时系统是指能够实现这样功能的计算机系统：操作系统通过调配，能够允许多个程序同时运行，每个程序被分到一定比例的计算资源。这样首先可以提高计算资源的利用率¹，并且一些不太耗时的任务能够更快地得到反馈，而不用等待它前面的程序全部运行完毕；同时，交互操作可以通过中断（interrupt）的方式传递给操作系统，操作系统调配出一小片时间（time slice）来处理交互操作，处理完之后可以迅速回到原来的计算任务，这样的话交互所造成的性能损失也相对可以接受了 [1, p. 100]。

分时的直接结果就是允许多个人使用同一台计算机，每个人通过各自的终端于其进行交互，并且运行

¹例如某个程序需要更密集的计算，而另一个程序则需要更多的 I/O，将它们同时运行可以更充分地利用计算机的资源。

各自的程序，这个功能尤其受到了大学的欢迎，因为它们需要让许多学生使用同一台电脑来运行各自的程序，并且得到及时的反馈 [1, p. 203]。而在 70 年代中期，这一功给办公室带来了“自动化办公”（office automation）——通常一间办公室配备一台小型机，然后每个职员配备一台终端连接到这台机器上，CRT 显示器、键盘以及文字处理（word processing）软件开始取代纸笔、打字机和文件夹 [1, pp. 254-263]。

在批处理时代，人们对于计算机的唯一期待就是它能够过一段时候之后计算出需要的东西，正如书中 [1, p. 122] 指出：

它（批处理）使得计算机就像是不需要马的交通工具，或者是无线的电报——它比制表机或者手算更快、能处理更多的数据，但是没有改变工作的实质。

It made the computer the equivalent of a horseless carriage or wireless telegraph — it worked faster and handled greater quantities than tabulators or hand calculations, but it did not alter the nature of work.

但是分时系统不同，用户可以像对话一样，通过终端和计算机进行实时的交互。同时，计算机的小型化廉价化也使得它不再是大公司的专利，越来越多的小型组织机构和个人也可以购买和使用计算机。在这个过程中它也不再只被用于朴素意义上的“计算”，而是成为了一个通用的“工具”，一个可以对话的“主体”，人们可以为它创造出更多新的用途，将实体的信息数字化，传递给计算机进行处理。除了上文提到的数字化办公，人们甚至还可以将小型机安装到拖拉机上分拣土豆 [1, p. 136]。交互这个过程不再仅限于输入和输出的“放置”和“拿取”，而需要在系统层面系统将其作为从外部传入的信号（signal）

此外，值得一提的是，分时计算机的实现中最重要的特性就是操作系统这一层“抽象”：在操作系统的管理下，无论是计算机的计算资源还是内存资源、存储资源，都被抽象成某种虚拟的接口，只要遵循这些接口，在编写程序的时候，就可以当作是这个程序使用了计算机的全部资源，而不需要担心和其他程序的冲突 [5]。

3 人的数字化——个人计算机与网络

3.1 个人计算机的出现

个人计算机，也就是为了供个人使用而制造的计算机，作为我们最熟悉的计算机的形态，它和上文所述的大型机、小型机在许多方面表现出质的差别：体积、价格、性能、交互功能。为了考察这样一种新的计算机形态是如何出现的，首先可以注意到两个技术上的因素：

其中一个因素则是分时系统的成熟。从 60 年代末起，分时系统已经广受关注，许多大学也开始研发分时系统。例如 1972 年 MIT 为 DEC 公司的 PDP-10 计算机研发的 TOPS-10 操作系统不仅支持分时、多用户使用，还支持了我们所熟悉的“文件”的概念，以及在终端上修改文件的文本编辑器 [1, pp. 207-209]。人和计算机已经可以进行比较复杂的交互，现代操作系统已经初见雏形。

另一个因素是半导体行业的集成电路技术的发展。Gordon Moore，仙童半导体以及后来 Intel 的联合创始人，在 1964 年就观察到：自从集成电路出现以来，集成电路上的晶体管密度每年都会翻一倍¹，按照这个趋势，Moore 预言 70 年代中叶人们就能够以低廉的价格购买到晶体管数量相当于 50 年代大型机的集成电路。在 60 年代末，一种被称为 MOS 的半导体技术出现，相比起之前被广泛使用的 TTL 技术，它极大地提高了集成电路上的晶体管密度 [1, p. 217]。

但是这两个因素并没有直接催生个人计算机。事实上，由于当时计算机最主要的客户，即大型商业公司对于分时系统并没有表现出太大的兴趣，因此许多分时系统制造商面临财政上的困难；同时虽然半导体厂商们在 70 年代制造出了小型的集成电路芯片，即微处理器（microprocessor），但是半导体厂商只考虑到将其用于计算器和工业用的嵌入式控制器 [1, pp. 221-223]。

个人计算机最终的出现，技术爱好者（hobbyist）群体发挥了关键性的作用。这样一个群体主要由活跃的无线电爱好者组成，他们爱好捣鼓机械和电路，但是常常没有足够的资金去购买当时市面上昂贵的计算

¹随着集成电路技术增速的放缓，这一定律后来也被人修改为每 18 个月或者两年翻一倍。

机。廉价的微处理器的出现引起了他们的注意，有一些爱好者购买这些芯片并且组装成可用的电脑，将其在面向这些爱好者的杂志上刊登广告售卖。早期的产品由于各种原因影响力都比较有限，直到 1975 年《流行电子》（Popular Electronics）杂志上出现了一则名为 Altair 的计算机的广告，宣称其“性能可以和当今的商业小型机抗衡”（performance competes with current commercial minicomputers）。而得益于使用 Intel 的 8080 微处理器，其售价不足 400 美元；相比起来，当时最廉价的小型机的售价也达到了数千美元 [1, pp. 224-230]。

虽然 Altair 实际上的功能非常有限，但是由于其低廉的价格和便于扩展的架构受到了爱好者的欢迎。和之前任何计算机不同，个人电脑在创造之初，最受欢迎的用途并不是任何正式的工作或者计算任务，而是电子游戏——事实上当时并没有人将它当作一个严肃的工具，自然也就缺乏进行这些工作的软件。但很快 Altair 的大受欢迎就催生出了许多用户群体（user groups）、报刊杂志、俱乐部，一些商业公司和爱好者组成的小公司也开始给 Altair 开发配件 [1, p. 231]。其中，当时还是 Harvard 学生的 Bill Gates 和他的朋友 Paul Allen 在 1975 创立的 Microsoft 为 BASIC 编程语言开发了性能优秀的编译器，从而补充了 Altair 在编程工具上的不足 [1, pp. 232-235]。廉价、可随机读写的软盘（floppy drive）在 70 年代后半叶的流行也使得 Altair 有了可靠的存储工具 [1, pp. 232, 236]。

几年之后，个人计算机的市场已经相当繁荣了。1977 年 Apple 发布的 Apple II 取得了成功，它卓越的图形功能不仅使得它适合于更刺激的电子游戏，也催生了 1979 年极其流行的 VisiCalc 电子表格软件 [1, pp. 264-266]。Apple 在 1984 年推出的 Macintosh 更是个人计算机带来了真正的窗口化的图形界面 [1, p. 273]。而 IBM 也在 1981 年进入了个人计算机市场并一如既往地占据大量市场份额 [1, pp. 268-271]。在这个过程中，Microsoft 依靠向个人计算机制造商出售 BASIC 和 MS-DOS 操作系统逐渐成长起来 [1, p. 271]。90 年代，Microsoft 推出的 Windows 系列操作系统取得了成功，并成为了个人计算机市场的主流 [1, p. 276]。

3.2 互联网与万维网

1967 年，美国国防部设立了高级研究计划署（Advanced Research Project Agency, ARPA），随后这个研究机构开始讨论如何将不同地域的电脑连接起来，以共享不同电脑的资源。这个被称为 ARPANET 的实验性的项目很快就被建设起来，到 1971 年，已经有 17 个节点上的 23 台计算机被连接起来了 [1, p. 194]。1981 年，美国国家科学基金会（National Science Foundation）建立了计算机科学网（Computer Science Network）使得研究机构可以接入 ARPANET，而 1986 年建立的国家科学基金会网络（National Science Foundation Network）则使得研究者们可以远程访问和使用超级计算机（supercomputer），这个网络逐渐蔓延到欧洲、澳洲、日本等地区。90 年代初，商业公司开始接管这些网络设施并且将它们向公众开放，全球、公开的互联网（Internet）互联网系统得以建立起来。早期的互联网通常是通过电话线进行通信的 [1, pp. 295-296]。

而在 ARPANET 以外，还有一些民间的网络通过电话线建立起来。例如 70 年代末一些爱好者建立的非常简单的计算机通信网络——业余爱好者网（FIDONET），在其基础上的电子布告栏系统（Bulletin-Board System, BBS）在这些群体之间颇为流行。许多大学内部也建立起了这样的小型网络 [6, pp. 11-18]。

互联网出现的另一条线索则要追溯到 UNIX。在 70 年代，Bell 实验室的 Ken Thompson 和 Dennis Ritchie 开发的一套称为 UNIX 的用来处理文件读写的程序，这一套程序组成了一个操作系统的雏形。由于 AT&T 将 UNIX 系统的源代码以几乎免费的价格向外分发，并且它可以在任何包含 C 编译器的平台上运行，因此 UNIX 在大学中十分受欢迎，教师和学生可以自由修改 UNIX 的代码以满足自己的需求。而 Berkeley 正是 ARPANET 的节点之一，学生们在改进 UNIX 的过程中将校内的网络和 ARPANET 融合在一起，其用户群体也逐渐广泛起来。1980 年 ARPA 将 Berkeley 学生们修改的 UNIX 版本（称为 BSD）作为 ARPANET 中推荐使用的操作系统，并且为其增加了 TCP/IP 网络协议的支持。1982 年 TCP/IP 的规范确立，并且成为 ARPANET 的标准，它随后也将成为互联网的基础协议。在这个过程中 UNIX 也和网络紧紧联系在了一起，UNIX 的用户也成为了互联网的最早期的一批用户 [1, pp. 281-285]。

互联网为全球的人提供了一个传输数据的通道，尽管其从 ARPANET 中继承下来的目的只是登陆远程电脑和传输文件，但是人们很快就为其发明了更多的使用方法，包括电子邮件，BBS 等。而让互联网真正成为一个包罗万象的平台的，则要归功于万维网（World Wide Web, WWW）。万维网的基本理念便是把

信息用一种统一的超文本标记语言（HTML）表示（也就是我们熟悉的网页），并且通过通用资源定位符（URL）标记它在网络中的位置，这样人们可以通过在浏览器（browser）中输入 URL 来访问对应的资源；其中超文本传输协议（HTTP）用于这个过程中的网络通信。一切资源都可以用一个 URL 表示，通过超链接（hyperlink）的方式可以任意在不同资源之间跳转，这样的思想构成了万维网的核心。Berners-Lee 在 1989 年开发了万维网并将它在欧洲核子研究中心（CERN）内部的网络使用，1991 年这套技术被开放给公众 [1, pp. 301-302]。

3.3 互联网时代

在本书的末尾（截至 2003 年），作者见证了对 Microsoft 的反垄断诉讼，dotcom 浪潮中门户网站（如 Yahoo）、搜索引擎（如 Google）、购物网站（如 Amazon, eBay）的出现，自由软件运动和开源软件运动的兴起 [1, pp. 308-325]。得益于时代，我们可以“预测”这个时间点之后计算领域的一些变化：便携计算机（如笔记本电脑）成为个人计算机中的重要组成部分，社交媒体和即时通讯工具已经无处不在，移动互联网和移动设备（手机、平板电脑等）使得它的用户群体和使用场景空前广阔，人工智能已经能够解决大量过去被认为只有人才能解决的问题，开源软件社区的力量逐渐庞大起来并且越来越受到商业公司的关注。越来越多的人发现这些设备和互联网成为了生活中不能缺少的部分。

如果说小型机和分时系统改变了人和计算机的交互方式的话，那个人电脑和互联网则是改变了人与计算机的关系。直到个人计算机出现之前，计算机都只是作为一个“工具”存在，无论是直接将纸带塞进它的输入口，还是用终端输入指令，计算机的价值都是快速、准确地完成某个工作。但是个人计算机，它起源于爱好者用来捣鼓的玩具，而这种玩具的属性也一直继承下来。一个人使用个人计算机（包括其衍生出的移动设备，下同），其目的不再和某个具体的任务联系在一起，而可能是纯粹的为了使用计算机里的某些功能（游戏、写作、绘图）。计算机的运行越来越“完好”，它也逐渐不再是一种纯粹的媒介，其本身已经成为了某种目的。

而互联网则让这个过程走得更远。而当互联网出现之后，它开始获得某种独立于现实世界的特征：一方面它是开放的，大部分的互联网信息都可以无障碍地、在地球上任何地方、以任何符合网络协议的方式获取，至少对于早期的互联网是这样的；另一方面，它是匿名的，对于某个互联网上的用户，人们能够直接获取的信息也许只有其用户名，一个人可能在互联网上拥有许多身份，甚至多个人也可以在互联网上拥有同一个身份，而互联网上的虚拟身份和现实身份的对应关系常常是难以确立的，虚拟身份也因此可以作为一个独立的主体存在（受到监管的压力，互联网的匿名性必然会逐渐降低，但是很难完全消除）。因此，从根源上讲，互联网构成了一个匿名的、开放的、全球化、去中心的公共空间。计算机成为了人与这个空间之间发生联系的入口（gateway）。

而这个入口的钥匙，或者说我们维持虚拟身份的方式，基本上可以归于我们大脑中的一串代表密码的字符，或者是插在手机里面的 SIM 卡，以及在服务器数据库中储存的对应的一串 0 和 1。无论如何，数字都已经成为了我们在互联网空间的代理人，而这个空间与现实的距离正越来越近。除了我们自身以外，我们与外界的关联也逐渐沉入数字化的空间：电子邮件，即时通讯工具，电子商务，电子书，视频会议，电子证件……没有任何迹象表明这个清单会在某个时刻停止增长。

4 结论

从过程来看，这段“计算的历史”当中，半导体技术的进步一直是最基础的驱动力，无论是哪一次新技术的出现，都能看到半导体行业的影子。但是，半导体也从来不是唯一的驱动力。冷战背景下美国军方对于新技术的狂热追求，企业和政府部门越来越多的数据的处理需求，繁荣的风险投资市场给创业者提供的机会，大学里面的计算机科学家们和以捣鼓新技术为荣的学生们，热衷于自己动手的无线电爱好者们，这些杂多的社会因素都或多或少地推动了历史进程。

而在这些书中所讲述的这些真切影响到历史进程的群体之外，我们也可以回顾一下理论家们曾经对计算机未来的设想。早在 50 年代，Douglas Engelbart 就认为计算机应当是增强人类智慧（augment human intellect）和构筑集体智慧（build collective intellect）的工具 [7]，他始终致力寻找实现这一目标的手段，除

了鼠标 (mouse) 这一广为人知的发明之外, 他在 1968 年所展示¹的一个包含图形界面、远程协作、视频会议的概念计算机系统也表现出了其惊人的前瞻性。Alan Kay 认为计算机就像印刷术一样, 是一种新的交流的媒介 (medium) [8]。Steve Jobs 将计算机比作思维的自行车 (bicycles for the mind), 能够突破思维的物理局限 [9]。

从结果来看, 计算机它们已经渗透入了生活并且成为了无处不在的基础设施 (infrastructure)。过去的半年已经证实了即使是像教学这样的活动, 也能在可以忍受的代价下完全通过互联网进行。某个事物一旦数字化, 计算机就能以远超人类的高效率对其进行处理, 互联网就能将其以快速、准确地将其分发。稳定、高效、可扩展 (scalable) 满足了工业革命后人们对于技术的想象。而另一方面, 数字化的信息能够以可以忽略的成本进行复制, 而互联网则为人们提供了一个只需要对彼此最小程度的信息和最低限度的成本就能展开交流的平台, 因此它又向人们展示了一个平等、自由、开放、去中心化的一个数字乌托邦 (Digital Utopia) 的新想象。热衷于技术的 hacker 们, 反对版权限制的自由软件运动的拥趸者们 [10], 很难说没有受到这样一种想象的影响。

但是当这样一种想象真正嵌入到社会时, 它又不可避免地要被商业社会、政治需求和社会结构所重塑。正如书中所提到的 [1, p. 349]: “UNIVAC 没有解决失业的难题, 个人电脑也并没有让普通人和有权势之人能够平起平坐”。Apple 在 1984 年的著名广告²中暗示 Macintosh 的创造力和自由能够使得这一年不会成为 George Orwell 在小说中预见的《1984》, 但颇为讽刺的是, 几十年之后人们开始察觉到, 某种程度上计算机和互联网所组成的无处不在的数字体系和《1984》呈现出越来越多的相似之处。

参考文献

- [1] CERUZZI P E. A History of Modern Computing[M]. 2nd ed. London, Eng. ; Cambridge, Mass: MIT Press, 2003. 445 pp.
- [2] Da CRUZ F. The IBM CPC[EB/OL]. 2003 [2020-06-16]. <http://www.columbia.edu/cu/computinghistory/cpc.html>.
- [3] LAMPSON B. Principles for Computer System Design[EB/OL]. [2020-06-18]. <https://bwlampson.site/Slides/TuringLecture.htm>.
- [4] IBM ARCHIVES: DPD Chronology - Page 3[EB/OL]. (2003-01-23) [2020-06-18]. http://www.ibm.com/ibm/history/exhibits/dpd50/dpd50_chronology3.html.
- [5] ARPACI-DUSSEAU R H, ARPACI-DUSSEAU A C. Operating Systems: Three Easy Pieces[M/OL]. [S.l.]: CreateSpace Independent Publishing Platform, 2018. <http://pages.cs.wisc.edu/~remzi/OSTEP/>.
- [6] CASTELLS M. The Internet Galaxy: Reflections on the Internet, Business, and Society[M]. Reprinted. Oxford: Oxford Univ. Press, 2003. 292 pp.
- [7] MARKOFF J. An Interview with Douglas Engelbart. 2002 [2020-06-21]. <https://youtu.be/VeSgaJt27PM>.
- [8] BARNES S B. Alan Kay: Transforming the Computer into a Communication Medium[J/OL]. IEEE Annals of the History of Computing, 2007, 29(2): 18-30 [2020-06-21]. <http://ieeexplore.ieee.org/document/4287214/>. DOI: 10.1109/MAHC.2007.17.
- [9] KOTTKE J. Steve Jobs and “the Bicycle for the Mind” [EB/OL]. (2010-10-15) [2020-06-21]. <https://kottke.org/10/10/steve-jobs-and-the-bicycle-for-the-mind>.
- [10] STALLMAN R, GAY J. Free Software, Free Society: Selected Essays of Richard M. Stallman[M/OL]. 1. ed. Boston: SoHo books, 2010. <https://www.fsf.org/bulletin/2010/fall/free-software-free-society-second-edition>. 224 pp.

¹见 <http://1968demo.org/>。

²见 <https://youtu.be/VtvjbmoDx-I>。