Nervous System #28: Triumphs of a Lazy Programmer

BY DAVID KALAT

With the aggressive pace of technological change and the onslaught of news regarding data breaches, cyber-attacks, and technological threats to privacy and security, it is easy to assume these are fundamentally new threats. The pace of technological change is slower than it feels, and many seemingly new categories of threats have been with us longer than we remember.

Nervous System *is a bimonthly blog that approaches issues of data privacy and cyber security from the context of history—to look to the past for clues about how to interpret the present and prepare for the future.*

This month marks the anniversary of the publication of Grace Hopper's landmark work in computer science, "The Education of a Computer." In seven typewritten pages published by the Remington Rand corporation in 1952, Hopper bequeathed to future generations something computer scientists in her day resisted, but programmers today likely take for granted: the ability to write code in programming languages that are (reasonably) understandable to humans. This was not always the case. In fact, in Hopper's day, being a computer programmer meant physically connecting wires and switches at a level that today's coders would consider engineering, not programming.

Hopper, a US Navy rear admiral and Ph.D. in mathematics, is one of the most important figures in computer science. She famously had a specially made wall clock that ran counterclockwise as a constant reminder to *"think differently."* She was born at a propitious moment in history. She received her degrees from Vassar in 1928 and Yale University in 1930 and 1934 at a time when women were receiving doctorates at a rate not matched until the 1980s. The onset of World War II opened job opportunities that had previously been denied to women.

Hopper's first attempt to enlist in the war effort in the immediate aftermath of the attack on Pearl Harbor was rebuffed due to her age and small size, but still she persisted and joined the US Navy Reserve. She was assigned to work on a project at Harvard designing computers to solve the complex calculations for weapons systems to fire accurately at airplanes. She was one of the programmers working on the Harvard Mark 1 computer—a fifty-foot-long collection of wheels, gears, and switches cabled together by 530 miles of wires. To instruct the computer meant first working out which switches should be switched in which way, and which bits needed to be plugged into other bits. Then she had to personally flip those switches and connect the wires.

In 1949, Hopper joined J. Presper Eckert and John Mauchly at the Eckert-Mauchly Computer Corporation to help them build computers for commercial purposes. Once again she confronted the mind-numbing aspects of programming early computers.

In the spirit of "exasperation is the mother of invention," Hopper's insight came from recognizing how much of her time and attention were spent hardwiring the same routines, over and over. Certain low-level tasks tended to repeat in different algorithms. She cataloged these "bread and butter" routines and assigned them symbolic codes.

Hopper could then string together more easily a sequence of these symbolic codes and let the "compiler" do the drudge work of translating the steps into the corresponding machine language. This permitted her to, as she put it in her landmark paper, "return to being a mathematician." This brilliant method of reusing software code has since become so familiar as to pass almost without notice. As Hopper quipped, "No-one thought of that earlier, because they weren't as lazy as I was."

Hopper continued to improve her compiler programs, with the ultimate goal of enabling programmers to write code in ordinary English, while a back-end of compilers would convert that human language into machine language. That goal remained for her—and for us—elusive, but it animated her efforts.

Happy users of her compiler started mailing her their reusable routines, which she added to the compiler library for the nest release. This process evolved into one of the first computer programming languages, the Common Business-Oriented Language (COBOL). COBOL emerged in 1959, and it too proved to be a lasting influence on the world of computers.

COBOL became the most commonly used programming language for business applications in the middle of the twentieth century—so ubiquitous, in fact, that many organizations and governments never replaced that code. This surfaced unexpectedly in the present-day COVID-19 pandemic, when state governments around the United States found themselves overwhelmed by coronavirus-related unemployment claims. These governments still ran their systems on COBOL, which created an unprecedented need for competent COBOL programmers to help improve and update the programs to process the claims.

Although Hopper shrugged off her invention of software compilers as something done out of "laziness" so she could avoid the drudgery of machine coding, it was a tremendous leap forward into a new age of software coding. Programmers today can thank Hopper that they do not need to hand-wire their programs into circuit boards.

Legaltech® news

This article was originally published in Legaltech News on May 5, 2020.