

Fernando J. Corbató 1926-2019
by Jerome H. Saltzer

Everyone who knew Fernando José Corbató called him Corby. A sure way to know when someone was a stranger was when that person addressed Corby as "Fernando".

Corby passed away at age 93 on July 12, 2019, in Newburyport, Massachusetts. He was born July 1, 1926, in Oakland, California, where his parents, Hermenegildo and Charlotte (Jensen) Corbató, were students at UC Berkeley. The family moved to West Los Angeles when his father became a professor of Spanish literature at UCLA. When Corby turned 17 in the middle of World War II he enlisted in the Navy as an electronic technician. The experience of dealing with the many kinds of electronic equipment onboard ship kindled his interest in systems. After the war he enrolled at Cal Tech, graduating in physics in 1950, and moved to MIT for graduate studies. In 1956 he completed his doctoral thesis, "A calculation of the energy bands of the graphite crystal by means of the tight-binding method", under the supervision of John Slater.

That thesis involved extensive calculations, which led him to become intimately acquainted with MIT's recently acquired Whirlwind computer. In oral history interviews, Corby said that he found the computer to be more interesting than the physics, so after receiving his Ph.D. and publishing three papers on the mathematics of computing he turned his attention toward the design and implementation of computer systems.

He joined the faculty of the MIT Department of Electrical Engineering in 1962 and was chair of Computer Science Curriculum Development until 1974, when the department added "Computer Science" to its name and he became Associate Department Head for Computer Science, a role he held until 1978 and again from 1983 to 1993. As a faculty member he taught MIT computer science subjects and served on a long list of Institute and departmental committees before retiring as Ford Professor of Engineering in 1997.

Corby developed MIT subject 6.251, Digital Computer Programming Systems, which was noted for one of his innovations: in most computer programming classes, students were expected to practice writing a few programs, but computer time and resource limitations meant that those programs could only be toys. In 6.251, students were handed the code for a complete working computer systems program (a two-pass assembler) and asked to modify it by adding features. Reading code gave insight into how systems programs were structured and adding features provided real-world experience in changing (and then debugging) complex programs.

In addition to his faculty roles he was Assistant, Associate, and then Deputy Director of the MIT Computation Center from 1959 to 1966. He was a founding member of MIT Project MAC (now the MIT Computer Science and Artificial Intelligence Laboratory), where he was Group Leader for Computing Systems Research from 1963 to 1974. From 1970 to 1973 he was a member of the Computer Science and Engineering Board of the National Academy of Sciences and in 1974 he was an ACM National Lecturer. From 1980 to 1983 he took on an assignment as MIT Director of Computing and Telecommunication Resources, bringing order to the proliferation of local computer networks that were springing up around MIT. (In that role he was colloquially described as "network czar".)

Corby is best known for his work on time-sharing. Using Whirlwind, in effect an expensive personal computer for which one signed up for brief blocks of time, he had experienced hands-on, interactive computing. In contrast, MIT's IBM 704 computer shared the cost of the expensive hardware by automating the running of a batch of computing jobs of several users, thereby sacrificing any opportunity for interaction--answers came back in hours or the next day. MIT faculty colleague John McCarthy articulated the goal of sharing the hardware cost while simultaneously providing interactive response and he demonstrated the attachment to the IBM 704 of a Friden Flexowriter controlling a program that ran between batch jobs. Faculty colleague Herbert Teager undertook to expand that experiment with ambitions to extend its interactive capabilities and to allow multiple terminals but did not make much progress. Corby was the one who set out to implement a complete system: hardware, software, operating staff, maintenance team, and

communications. He acquired the resources, developed key innovations, and personally did much of the programming. Two innovations were short-term scheduling of the hardware with a multi-level feedback queue, which gave each user the illusion of a private, hands-on computer, and on-line storage for both programs and data, which allowed a user to take a break from interactive work, grab lunch, and continue when convenient, minimizing shut-down and start-up time.

The result, demonstrated in November 1961, was the Compatible Time-Sharing System (CTSS) for MIT's newly-installed IBM 709 computer. Within a year he and a small staff of dedicated programmers escalated CTSS into a production system that could handle up to 32 terminals, disk storage, and a private dial-in branch exchange. In the summer of 1963 CTSS hosted a summer study in which hundreds of knowledgeable computer folks from around the world came to try out time-sharing.

Seeing how users reacted to the availability of on-line storage led Corby to realize that another goal is also important: the ability for users to share information, both programs and data. At the same time he foresaw that information sharing raised concerns about privacy and security and he added controls on the extent of sharing in the form of user names, passwords, and permissions. Having established the value of information sharing, Corby went on to lead the design of a follow-on system, Multics, which was a rethinking of time-sharing with information sharing at the center. Where CTSS was a time-sharing system, Multics was a computer utility from which a user could obtain computing "by the yard" and share anything.

While leading the technical design of these two systems, Corby made several recognized contributions to the management of software development. The dominant paradigm for developing software in the 1960s was to gather requirements, design a system, implement the designed system, and deliver it to users, an approach later named the Waterfall Model. Corby realized that novel software systems need a more iterative approach, and he pioneered several software management practices. Probably the most important of these was to first build a minimal system that does something useful. Then, once it is working, add features one at a time by revision, at each point checking to see that the system still works. Although delay in hardware delivery forced development of Multics to start with a more Waterfall-like model, once the system was operable the Multics team switched to the iterative revise-as-you-go method.

The development of both systems exploited a second innovation: As soon as possible the system being developed became the system used for its own development, a practice that a few years later would become known as "eating your own dog food". A third innovation was that Multics was probably the first large operating system that was recompiled from scratch every night, installed, and used to continue system development the next day, thereby quickly revealing bugs in implementation, usability, and usefulness. Corby also encouraged intensive communication among the staff, with documentation before implementation, design reviews, and frequent staff-wide meetings in which project members explained what they were doing. His model was that everyone working on a complex project should be aware of what everyone else was doing.

Decades later, these management practices would be labeled with the buzzwords "agile" and "DevOps".

In another influential move, Corby expanded on the Burroughs Corporation innovation of using a high level language for system programming. This technique was applied minimally in CTSS, writing the scheduling algorithm in the MAD language to make it easy to understand, change, and publish. In the follow-on, nearly all of Multics was written in a version of PL/I, adopting Corby's observation that the number of lines of code that a programmer can produce and debug in a day is independent of the language, so a higher-level language that allows programs to be written in fewer lines provides a significant advantage. This anecdotal observation has enough truth to it that others have dubbed it "Corbató's Law".

Corby published papers on system design and software management but his larger influence on the world has been via students, staff, and colleagues who developed and used CTSS and Multics and then went out and incorporated the ideas in their own

system designs. For example, IBM's CP/CMS and its successor VM/370 were developed by staff who helped implement CTSS. Similarly, the Unix system was developed by staff who helped implement Multics. Students in Corby's research group went on to develop electronic spreadsheets for personal computers, develop operating systems for several computer industry startups, and lead development of the architecture of the Internet. Today's computing cloud is directly traceable to the ideas of the computer utility and computing by the yard that inspired both CTSS and Multics, but which could be only partly realized with the hardware and communications available 50 years ago.

Corby's list of awards and elections to the rank of fellow attests to the influence his work has had: IEEE W. Wallace McDowell award (1966); IEEE Fellow (1975); AAAS/1780 Fellow (1975); NAE election (1976); AFIPS Harry Goode Memorial Award (1980); IEEE Computer Society Pioneer Award (1980); AAAS/1848 Fellow (1982); ACM Computer Pioneer Charter Award (1982); ACM Alan M. Turing award (1990); ACM Fellow (1994); NEC C&C prize (1998); Computer History Museum Fellow (2012).

Finally, Corby's character was modest, gentle, civil, and personally thoughtful, an engaging human being. A few anecdotes and personal notes illustrate.

Former staff member Tom VanVleck explained why he enjoyed working with Corby: "Everyone on his staff was confident that Corby could do their job if necessary. He understood what they were doing and how they were doing it and could help solve their problems."

No matter how off-the-wall someone's proposal was, Corby always listened calmly and responded gently. At the same time he had an uncanny ability to come up with just the right comment that explained how off-the-wall the idea was. Similarly, he was a master of elliptic criticism. Rather than saying "You did it wrong!" he would say "Sometimes people make the mistake of...", and then go on to describe the kind of mistake you didn't realize that you just made.

A typical piece of advice from Corby that illustrates his way of thinking about the world as a system: To keep your office from overflowing, look at your in-basket each morning and use that as a guide to the amount of stuff you should throw away today. (This advice clearly preceded the e-mail inbox.)

Another typical piece of Corby's advice: Updates for a loose-leaf notebook should pass the drop test. If you drop the notebook and its update pages on the floor, there should be enough information on each sheet to reassemble the notebook in the proper order and correctly finish updating it.

Although Corby and his second wife Emily had four children and his house in Newton, Massachusetts, had a back yard with room to add a swimming pool, to avoid the temptation Corby kept handy a catalog of swimming pool equipment and accessories to remind him why he should not.

Corby and Emily were gracious hosts of dinners in Newton and summer visits to their getaway cottage on Newburyport's Plum Island. The conversations were always wide-ranging yet without getting into professional topics.

Corby was predeceased by his parents and his first wife, Isabel Blandford (1933-1973). He is survived by their two daughters, Carolyn Suzanne Corbató Stone and Nancy Patricia Corbató; a brother, retired professor of Geophysics at The Ohio State University, Charles Corbató; his second wife, pianist and photographer Emily Susan Gluck; and two stepsons, Jason Charles Gish and David Edward Gish. The New York Times obituary by Katie Hafner reports five grandchildren.

Adapted and expanded from a talk given at a memorial service at MIT on 4 November 2019.