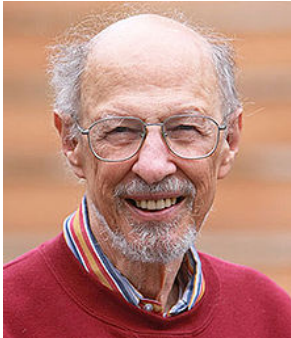


# Fernando Jose Corbató

*Born July 1, 1926, Oakland, Calif.; creator of the Compatible Time-Sharing System (CTSS), the first general-purpose interactive system.*



*Education:* BS, physics, California Institute of Technology, 1950; PhD, physics, Massachusetts Institute of Technology, 1956.

*Professional Experience:* US Navy, electronics technician, 1943-1945; research associate, MIT Computation Center, 1956-1959; assistant director for programming research, 1959-1960; associate director, 1960-1963; deputy director, 1963-1966; group leader in the computing systems research group, Laboratory for Computer Science, 1963-1972; co-head, systems research division, 1972-1974; co-head, automatic programming division, 1972-1974; associate head, Department of Computer Science and Engineering, 1974-1978, 1983-present; Cecil H. Green Professorship, 1978-1980.

*Honors and Awards:* W.W. McDowell Award, IEEE, 1966; Harry Goode Memorial Award, American Federation of Information Processing Societies, 1980; IEEE Computer Society Pioneer Award, 1980; ACM Turing Award, 1990.

When the war broke out in 1941 (for the US) Corbató's high school went into long hours and he saw a chance to get out in a hurry. He went to UCIA as a student, with the threat of the draft looming over his head. People came by the school who were concerned about the ability of the Navy to maintain and repair the incredible amount of electronic equipment they were getting. There was a program called the Eddy Program, and they gave Corbató the opportunity to join the program and get an education as an electronic technician. One of the benefits, of course, was one did not get drafted, or get assigned to be a cook, or something worse. So Corbató enlisted at the age of 17 in the Navy and went through a year-long program as an electronic technician. He got exposed to some of the earliest and largest electronic systems then deployed in the Navy—mostly radar, loran, and sonar systems—both on land and on ship. He did not realize until later how important that was.

After the war he got a chance to go back to college, this time at Cal Tech. Since everyone wanted to be a physicist in those days, Corbató read to be a physicist following his undergraduate studies with graduate work at MIT, still with the intention of undertaking a career as a physicist. He got his doctorate in physics, but along the way got exposed to digital computers by the late Philip M. Morse [later to be director of the MIT Computation Center]. In 1951 Morse recruited Corbató to take on a new research assistantship in the use of digital computers. The people who signed on with this initial ONR-sponsored research assistantship program were exposed to punched cards in the spring, and in the summer they were introduced to Whirlwind. The Whirlwind was just barely operational; it had approximately 1,000 words of 16-bit custom-built electrostatic memory that had a mean time between failures (on a good day) of about 20 minutes. Jay Forrester was very prescient in recognizing that the Williams tubes (Manchester University) were flaky in design, and he had actually gone back basically to first principles to design a very elaborate mosaic storage tube. Somewhat similar in storage retention (charge retention) to Williams tubes, it was a tour de force in electronics to make it work. A group headed by Patrick Youtz built them in a basement at MIT.<sup>1</sup>

The people who operated Whirlwind recognized the need to have unclassified work and be able to use it in an open way. So the deal they worked out was that approximately three to four hours a day were made available

---

<sup>1</sup> Wildes and Lingren 1986.

for that purpose. In the morning were the times when the non-classified personnel were permitted to use the machine. This meant that general users worked all day and all evening preparing Flexowriter tapes, which were the input medium, and then took a quick shot at the machine the next morning. If one were really fast, the user might get two shots in during each two-hour period. Most times it was one shot a day, which could be very frustrating, but on the other hand, the machine was not that big, and the programs were not that huge.

It was a personal computer; there were many things about it that got lost in the next few years, such as graphical display output and even audio output—there was a probe put on the circuit of the accumulator which gave the users a “signature” of each program they were running. The users actually could hear where the program was, could sense the loops, and could even get the tempo of how the program was running. Occasionally people recognized from the audio that they had a program that was misbehaving, because it did not sound as they thought it should at that point.

After Corbató got his doctorate in physics in 1956, Phil Morse recruited him into the newly formed Computation Center, which he had established that year. The focal point of the new center was an IBM-704; Morse had worked out a package deal with IBM, where in return for the use of the machine on campus for one shift, MIT would also make it available to a cooperative group of New England colleges for the second shift. The third shift would be retained to be used by IBM's own local scientific office. Time was also provided for a group in Cambridge, formed and directed by Martin Greenberger, that computed the orbits for the first USSR Sputnik.

Corbató played various roles in the center, ranging from initially supervising a research assistantship program to later becoming an assistant director, associate director, and finally deputy director of the Computation Center.

In the spring of 1961 when the IBM-704 had been replaced by an IBM-709, Corbató started up a project which eventually developed the Compatible Time-Sharing System (CTSS) with just a couple of the key staff people—Marjorie Daggett (who was then Margaret Merwin) and Bob Daley. By November 1961 they were able to demonstrate a really crude prototype of the system. They had eked out 5K words of the user address space from the standard operating system and inserted a tiny operating system that managed four typewriters. Backup storage was achieved by assigning one magnetic tape drive per typewriter.

This system could operate effectively as long as no other user wanted an “all-the-core-memory” type job to run under the Fortran (EAP) monitor system. This system could coexist with that kind of an operating system and could run jobs. So it could run compatibly; it could run while ordinary batch work was being run on the IBM-709. It used the same language systems as the batch system; thus, using CTSS meant not having to start programming over again, as had happened in most system and machine changes up to that time. Even early versions of LISP ran under CTSS.

The IBM-7090 replacement hardware arrived in the early spring of 1962, supporting the necessary interrupt capabilities which had not been available in the earlier machine. Corbató et al. presented a paper at the 1962 Spring Joint Computer Conference (Corbató et al. 1962), which suggested that they were running CTSS on IBM-7090 hardware on the basis that this would be fact by the time the paper was presented. In fact, it was not operational until the fall of that year.<sup>1</sup>

---

<sup>1</sup> Editor's note: A prime example of why historians cannot even always trust the primary sources of information on an event.

That paper was the first description of a working time-sharing system. It stated:

... it is best to give a ... precise interpretation to time-sharing. One can mean using different parts of the hardware at the same time for different tasks, or one can mean several persons making use of the computer at the same time. The first meaning, often called multiprogramming, is oriented toward hardware efficiency in the sense of attempting to attain complete utilization of all components. The second meaning of time-sharing, which is meant here, is primarily concerned with the efficiency of persons trying to use a computer. Computer efficiency should still be considered, but only in the perspective of the total system utility.

An experimental time-sharing system has been developed. This system was originally written for the IBM-709 but has been converted for use with the 7090 computer.

The 7090 of the MIT Computation Center has, in addition to three channels with 19 tape units, a fourth channel with the standard Direct Data Connection. Attached to the Direct Data Connection is a real-time equipment buffer and control rack designed and built under the direction of H. Teager and his group. This rack has a variety of devices attached but the only ones required by the present systems are three Flexowriter typewriters. Also installed on the 7090 are two special modifications (i.e., RPQ's): a standard 60-cycle accounting and interrupt clock, and a special mode which allows memory protection, dynamic relocation and trapping of all user attempts to initiate input-output instructions.

In the present system, the time-sharing occurs between four users, three of whom are on-line each at a typewriter in a foreground system, and a fourth passive user of the background FAP-MAD-MADTRAN-BSS Monitor System (FMS) used by most of the Center programmers and by many other 7090 installations.

Significant design features of the foreground system [for the user] are [that he can]:

1. Develop programs in languages compatible with the background system,
2. Develop a private file of programs,
3. Start debugging sessions at the state of the previous session, and
4. Set his own pace with little waste of computer time.

The foreground system is organized around commands that each user can give on his typewriter and the user's private program files which presently (for want of a disk [sic] unit) are kept on a separate magnetic tape for each user.

The commands are typed by the user to the time-sharing supervisor (not to his own program) and thus can be initiated at any time regardless of the particular user program in memory. For similar coordination reasons, the supervisor handles all input/output of the foreground system typewriters. Commands are composed of segments separated by vertical strokes; the first segment is the command name and the remaining segments are parameters pertinent to the command. Each segment consists of the last six characters typed (starting with an implicit six blanks) so that spacing is an easy way to correct a typing mistake. A carriage return is the signal which initiates action on the command. Whenever a command is received by the supervisor, "WAIT" is typed back followed by "READY" when the command is completed. (The computer responses are always in the opposite color from the

user's typing.) While typing, an incomplete command line may be ignored by the “quit” sequence of a code delete signal followed by a carriage return. Similarly after a command is initiated, it may be abandoned if a “quit” sequence is given. In addition, during unwanted command type-outs, the command and output may be terminated by pushing a special “stop output” button.

Although experience with the system to date is quite limited, first indications are that programmers would readily use such a system if it were generally available. It is useful to ask, now that there is some operating experience with the IBM-7090 system, what observations can be made. An immediate comment is that once a user gets accustomed to [immediate] computer response, delays of even a fraction of a minute are exasperatingly long, an effect analogous to conversing with a slow speaking person. Similarly, the requirement that a complete typewritten line rather than each character be the minimum unit of man-computer communication is an inhibiting factor in the sense that a press-to-talk radio-telephone conversation is more stilted than that of an ordinary telephone. Since maintaining a rapid computer response on a character-by-character basis requires at least a vestigial response program in core memory at all times, the straightforward solution within the present system is to have more core memory available. At the very least, an extra bank of memory for the time-sharing supervisor would ease compatibility problems with programs already written for 32,000-word IBM-7090's.

In conclusion, it is clear that contemporary computers and hardware are sufficient to allow moderate performance time-sharing for a limited number of users. There are several problems which can be solved by careful hardware design, but there are also a large number of intricate system programs that must be written before one has an adequate time-sharing system. An important aspect of any future timeshared computer is that until the system programming is completed, especially the critical time-sharing supervisor, the computer is completely worthless. Thus, it is essential for future system design and implementation that all aspects of time-sharing system problems be explored and understood in prototype form on present computers so that major advances in computer organization and usage can be made.

After the success of CTSS Corbató continued the development of time-sharing systems under Project MAC, which was headed by Robert Fano. While Project MAC initially used CTSS as its basic system, it was superseded by Multics operating on GE computers in the late 1960s, and Multics was marketed by Honeywell Information Systems for many years after GE sold its Computer Division in 1968.

In 1991 Corbató received the ACM Turing Award: “For his work in organizing the concepts and leading the development of the general purpose large-scale time-sharing system and resource-sharing computer systems CTSS and MULTICS.” His major research interests continue to be in time-sharing systems, automatic programming, and knowledge-based application systems.

## **QUOTATION**

Regarding the November 1961 primitive CTSS: “. . . we were just trying to get a demonstration system going to convince people that it was a good idea. A lot of people did not understand what it meant to interact. [That] was amazing.” (Lee and Rosin 1992)

## BIBLIOGRAPHY

### Biographical

- Corbató, Fernando J., “On Building Systems That Will Fail,” 1991 Turing Award Lecture, *Comm. ACM*, Vol. 34, No. 9, Sept. 1991, pp. 73-81.
- Frenkel, Karen A., “An Interview with Fernando Jose Corbató,” *Comm. ACM*, Vol. 34, No. 9, Sept. 1991, pp. 83-90.
- Lee, J.A.N., and Robert Rosin, “Time-Sharing and Interactive Computing at MIT,” Special Issue, *Ann. Hist. Comp.*, Vol. 14, Nos. 1 & 2, 1992.

### Significant Publications

- Corbató, Fernando J., et al., *The Compatible Time-Sharing System. A Programmer's Guide*, MIT Press, Cambridge, Mass., 1963.
- Corbató, Fernando J., et al., “Multics—The First Seven Years,” *Proc. Spring Joint Computer Conf.*, Vol. 40, AFIPS Press, Montvale, NJ., 1972, pp. 571-583.
- Corbató, Fernando J., “Time-sharing,” in Ralston, Anthony, and Edwin D. Reilly, Jr., eds., *Encyclopedia of Computer Science and Engineering*, 2nd ed., Van Nostrand Reinhold Co., New York, 1983, pp. 1520-1526.
- Corbató, Fernando J., Marjorie Merwin-Daggett, and Robert C. Daley, “An Experimental Time-Sharing System,” *Proc. Spring Joint Computer Conf.*, AFIPS, Vol. 21, The National Press, Palo Alto, Calif., 1962, pp. 335-344.
- Corbató, Fernando J., and V.A. Vyssotsky, “Introduction and Overview of the Multics System,” *Proc. Fall Joint Computer Conf.*, AFIPS, Vol. 27, Spartan Books Inc., Washington, D.C., 1965, pp. 185-196.

## UPDATES

Portrait changed (MRW, 2012)