Oral History of Fernando Corbató

Interviewed by:
Steven Webber

Recorded: February 1, 2006
West Newton, Massachusetts

CHM Reference number: X3438.2006

Steven Webber: Today the Computer History Museum Oral History Project is going to interview Fernando J. Corbató, known as Corby. Today is February 1 in 2006. We like to start at the very beginning on these interviews. Can you tell us something about your birth, your early days, where you were born, your parents, your family?

Fernando Corbató: Okay. That's going back a long ways of course. I was born in Oakland. My parents were graduate students at Berkeley and then about age 5 we moved down to West Los Angeles, Westwood, where I grew up [and spent] most of my early years. My father was a professor of Spanish literature at UCLA. I went to public schools there in West Los Angeles, [namely,] grammar school, junior high and [the high school called] University High. So I had a straightforward public school education. I guess the most significant thing to get to is that World War II began when I was in high school and that caused several things to happen. I'm meandering here. There's a little bit of a long story. Because of the wartime pressures on manpower, the high school went into early and late classes and I cleverly saw that I could get a chance to accelerate my progress. I ended up taking both early and late classes and graduating in two years instead of three and [thereby] got a chance to go to UCLA in 1943.

Webber: Let me back you up just a bit. Did you have any siblings, any brothers and sisters or—

Corbató: Yes. I have a brother. He's six years younger and he's had a nice, long career at Ohio State. He's a professor emeritus of geology. We're still close friends and in close contact.

Webber: During your elementary school and high school years, what types of things interested you? Were you in Scouts, were you in programs, did you do any music, those types of things?

Corbató: I think the thing that kind of got my interest most was mathematics and at the junior high school level that meant only algebra. I was good at that and enjoyed it. I was also good at the science classes and the like. So when I went to UCLA, I don't think I had a major yet, I was taking basically chemistry and physics classes and [similar courses]. My UCLA experience was cut short because about seven months into the first year a navy recruiter approached me and asked if I wanted to join a program that the navy was running called the Eddy Program. [The program was] named after a Commander Eddy, [and its purpose] was to train a cadre of electronic technicians for the navy. I was intrigued because it meant a year of schooling - in an area which sounded interesting. Significantly it meant that I could designate where I was going to go, namely the navy rather than the army, which was likely to be the case if I was drafted.

Webber: This was right in the middle of the war?

Corbató: This was in 1943, the war was already ongoing, and of course December 7th, 1941 was the beginning of our entry into the war. [The Eddy program] sounded pretty good and I impulsively left school and joined the navy.

I impulsively left school and joined the Eddy Program, of course with my parents' permission, and engaged in a year long program of training, starting with two weeks at the Great Lakes Naval Training Station and then another month in Chicago at a so called pre-radio school. Then [there were] three months out on the West Coast in El Monte, California, at what's now the Naval Postgraduate School in the old El Monte Hotel which the navy had commandeered. Then [there were] six months up at Treasure Island, midway between San Francisco and Oakland in the bay, in the final schools. The navy was trying to train people to service, maintain and debug the incredible array of new electronic systems which were being deployed into the fleet. [In particular,] sonar, radar, loran, depth finders, along with traditional radio and the like. An incredible array of equipment which was being foisted on the ships. They had no one to maintain them and they were trying to train a relatively sophisticated group of technicians to keep this equipment going. So that [was] my first year in the navy and then the second year [I] was assigned to a ship and seeing real world experience.

The most important thing about that experience was that it had a tremendous influence on the rest of my life in the sense that it made me think about systems and debugging, trying to divide and conquer, tracking down errors and that sort of thing. Then when I came out of the navy in 1946, I was able to afford to go to Cal Tech because the GI Bill was the law of the land for returning veterans. That was a tremendous help because I didn't have to hit on my family to pay tuition. So I went to Cal Tech for four years as an undergraduate, had a great time and majored in physics because everyone majored in physics in those days. It was a good, challenging undergraduate experience and then I started applying for graduate school. MIT was one of my choices. I was accepted to MIT and came back in 1950 to begin my physics career as a graduate student.

Webber: Did you not want to stay at Cal Tech or did you want just more diversity?

Corbató: Cal Tech had a pattern of not normally taking their own undergraduates. I think they would break it if they had a brilliant student --I was a good student but not a brilliant student-- so I clearly had to leave At MIT I came under the wing of Professor Phil Morse who was a physicist. He was quite entrepreneurial and had recognized early that computation was a coming thing.

Webber: Are you at MIT as a student now or—

Corbató: A graduate student in the physics department. I was also a TA in the physics department. But the key thing that Morse did was he drummed up the money for some research assistantships in computation. Remember computation was in a pretty primitive state in those days. To most people that meant a few exotic machines [somewhere] in the country like EDSAC and JOHNNIAC maybe and MIT's Whirlwind which was just being completed in 1950. Anyway, one of these research assistantships gave me a chance to get acquainted with everything from punched card equipment to actually spending some time learning to use Whirlwind. Whirlwind was a unique computer in [that] it was basically one of the first computers that you would call a RISC computer in the sense that it had a very simple instruction set. There were 32 instructions or order codes, a 16 bit word, a 24 microsecond memory cycle, and it was a parallel computer. It was very fast for its day because the original intent was for it to be a real time computer. It had been started probably around 1946. 1947 [was] when they began to build it, but it was only coming on line about 1950. The original intent had been for it to be an aircraft simulator, to essentially be programmed as an aircraft simulator, and so that's why it was parallel and fast. Jay Forrester at MIT was the lead figure in creating the machine. Anyway, I got a chance to learn how to use a digital computer and program it. It was a very pivotal experience because I became familiar with what we now consider modern computers.

Webber: What was the user interface? How did you actually use the computer?

Corbató: Good question. The computer input was basically from toggle switches which were only used to boot it up. I never did that. Or you could input to it from paper tape. There was no keyboard to my recollection that you could use to input to it. You had basically a paper tape reader. So we would prepare the paper tapes off line with machines called Flexowriters which would punch a paper tape that could be read in. It was a bit of a ritual. Basically you had some very elementary initial programs [as] loaders which would take the paper tape contents and fill memory up. Then you could start the program at some point.

Webber: Were there lights on this? Did it make noises?

Corbató: There were lights.

Webber: How did you know what was happening?

Corbató: You may or may not have been there when your program was run. My recollection is we would try to be there when it ran and that there were noises. The engineers had cleverly put an audio probe on one of the bits in the accumulator so a program made a signature racket whenever it ran and you could actually hear it. The tempo was such that you could decide if your program was in a loop. In other words, you had a signature of your program's operation and it was one form of debugging actually. If you heard the same sound in a steady drone, you were almost sure you were hung up in a loop.

Webber: That sounds like it would be useful today.

Corbató: Well, we lost that [feature]. In later years we tried to put that into the IBM 704 and 709 but IBM was not very interested. It was not part of their standard operating procedure. So yeah, it would be useful today.

Webber: You had good hands on experience with the Whirlwind. That was in the early 1950s while you were still a grad student. What other things happened during those grad student years?

Corbató: Well, I was working on my doctoral thesis [which] was doing energy band calculations in graphite, in carbon. It was very laborious type work and in retrospect kind of dull. But the thing that it taught me how to do was to write big programs, to debug them, to get a complete hands on feel for working with programs, organizing them, and learning the ropes of how to use a computer. Along the way, Professor John Little and I were both fellow graduate students, both in physics, both working for Phil Morse. He is now a professor emeritus in the Sloan School at MIT. He and I got interested in a project, I guess Morse probably suggested it, to create a numerical table of coefficients for expanding spheroidal wave functions. It was-- I'm losing track of the precise terminology --a table of numbers and this was comparable to [the work of] Aiken up at Harvard who was publishing Bessel functions. So we basically did a set of tables using Whirlwind to print this giant volume which was maybe a couple of inches thick, and which was nothing but numbers. That was a great tour de force and one of the earliest examples of trying to do automatic printing. In those days computing applications were usually done with hand calculators and having tables of coefficients printed was a useful adjunct. Today we would just generate them on the fly but in those days you had to archive them.

Webber: During your graduate school years, did you have any kind of a social life or were you always in front of the computer?

Corbató: Well, actually it did interact with my social life in the following sense. Whirlwind was being used as a pilot system for developing the SAGE system. We didn't know this because it was all under wraps. All we knew is that it was a classified activity. Well, we had a hint, but it was a classified activity and it meant that we only had access to the computer approximately three or four hours a day, [with] maybe an hour [from] 4 to 5 in the afternoon and then a couple hours from 2 to 4 in the middle of the night because the machine was being used by others. So that turned my life inside out. I was working nights and sleeping mornings. It taught me a little bit about the hazards of having crazy hours.

Webber: After your graduate school, I assume it was a degree in physics again?

Corbató: I got my doctorate in physics.

Webber: What happened next?

Corbató: Phil Morse recognized that Whirlwind was coming to the end of its useful life in the sense that the navy was going to stop supporting it because by then the SAGE system was under way and its need as a prototype for SAGE had disappeared. He had convinced IBM to put an IBM 704 computer at MIT as a computing center which he would operate for MIT. The deal was that MIT would have one shift of time; a second shift of time was to be devoted to supporting about 25 to 40 New England colleges which Phil Morse organized into a user consortium; the third shift was held by IBM; and there was a mysterious fourth shift which was the weekends which we sort of cadged. Anyway, this Computation Center got started about- I guess he cut the deal in 1956-- and right after getting my doctorate, I signed on as a research associate with his Computation Center. Phil Morse had also organized some more research assistantships and so one of my first responsibilities was to be the person in charge of supervising the research assistants that were associated with the new Computation Center. Anyway, I had a hand in the administration of the new Computation Center. I had access to the IBM 704 which was basically one of the earliest commercial scientific computers. Also as part of my responsibilities I began to go to SHARE meetings which were national meetings held twice a year by the customers of IBM who had IBM 704s. Now there must have been a few hundred of these machines scattered around the country, many of them in aircraft companies, doing the bread and butter engineering work. So I learned the state of the art of how computers were being used in those days.

Webber: Your first job after getting your doctorate was actually managing people and dealing with scheduling the computer center and working with those types of things or is it--

Corbató: Well, actually, I was on the side of that. I was initially handling part of the research adjunct of the center. There were other people, Frank Verzuh, in particular, who was the associate head of the center, and Phil Morse who was the director. Phil Morse would show up about one day a week and was always on call but he basically delegated the operation of the center to others. So Frank Verzuh was the person who was actually handling the administrative side of it initially. He left after a couple years and at that point Phil Morse asked me to take over that role. It was at that time that early timesharing began to bubble up.

Webber: That was the late 1950s then. How did that timesharing actually appear? What were some of the discussions?

Corbató: Well, let’s see. The center got operating around 1957 and in just a few short years we had developed an appetite for computing among the scientific and engineering community. [The demand] was pretty voracious and people were learning to program. My recollection is Fortran hit the world around 1956 and had a huge impact because it suddenly opened the door for people to not be coders but be real programmers in the sense that they could begin to deal in a semi algebraic form with their programs. So the demand on the Computation Center was ever increasing and growing by leaps and bounds. By 1958 and 1959 we were suffering from being overloaded. This was happening all across the country of course. The state of the art for the IBM 704 was initially for it to be driven by a deck of punched cards which you'd put in a punched card reader with output printed out on the online printer. People had realized that this was terribly inefficient and so they had gone to prerecording a batch of punched card jobs on magnetic tape, and then during the operation of a batch, the output stream would be recorded on magnetic tapes and subsequently printed off line afterwards. This came to be called batch processing and the obvious intent was to optimize and make the machine more efficient. But one of the consequences was that the individual programmer or user found himself with more and more frustrating access to the machine. It got to the point where under the pressure of people queuing up, you would get maybe one shot a day at running your program. Typically it might be just compiling it and trying to run it, but any error at all, a dropped comma, anything, would scrap the job and you would be left with a mound of paper which was the dump of memory, and trying to figure out what had gone wrong. So one run a day to find a trivial mistake was terribly frustrating and counterproductive.

In retrospect we came to recognize that what we'd done was we'd optimized the use of the machine but not the people. The first person to become really articulate about it that I heard was John McCarthy. John McCarthy had joined the Center when he joined MIT as a faculty member from-Dartmouth where he was a mathematics professor. He came down and became a professor of electrical engineering at MIT approximately in 1958. McCarthy was basically an early computer scientist and he had a lot of irons in the fire including developing the LISP language. But one of the things that he recognized too was that the computer was hard to get at and he proposed the notion of timesharing. It is too complicated to get into all the ins and outs of who came up first with the idea but John in my mind was probably the most influential person to articulate the vision of many people using a computer at once with a computer multiplexing between individuals at multiple terminals and having high interaction. In 1962, I think it was 1962, there was a book put out by Martin Greenberger which collected some invited lectures. In that book John articulated his ideas in print. Meanwhile orally he was promoting the job of doing timesharing. So he began to argue that we might be able to modify the IBM 704- excuse me by that time it had become a 709, a successor to the 704, but still a vacuum tube operated machine, and he proposed we start doctoring it to allow real time interrupts and to allow terminal access to the computer.

Webber: The real time interrupts are needed for time splicing? I'm surprised they weren't there before.

Corbató: Well, let's see. No. The machines were batch programmed and batch operated. There was no interaction with the jobs. That had been lost. The Whirlwind experience was down the tubes and IBM was trying to run a machine shop in effect. That was their pattern of all the users. What McCarthy was proposing was that we have some form of an interrupt timer which would allow the machine to be trapped back to a supervisory program which could then change the computer's operation to another user and do this in sort of a round robin fashion, to switch between users. There were a lot of other complications and McCarthy began to see them. One of the complications was that if you had many programs in memory at once, you had to keep avoiding having programs write over each other. If you could multiplex between them, how did you keep them from trampling over each other? So that brought in the notion of some sort of memory bounds registers. With an interrupt timer, and memory bounds registers, you needed the notion of being able to swap programs in and out of memory. This introduced the notion that you might not always want them to be in the same place in memory as they'd last been, so that introduced notions of relocation. So there were a lot of new complications that were suddenly creeping in.

Webber: Were you doing research on this or other projects at this time in your career?

Corbató: I was doing other things, writing programs for numerical computation, still doing some energy band stuff. I was in addition involved in a little bit of system programming in that I came up with a scheme, I think again McCarthy suggested it even, of a relocatable loader where you could load programs into different locations in memory from one time to another. It's far back at this point.

Webber: Those are so embedded in the new hardware that we have today. It's hard to imagine that was never there. How did you get involved actively in the timesharing effort yourself?

Corbató: Well, I feel we're rambling a little bit. Another person at the center was Professor Herb Teager. He had joined the EE faculty and got interested in the hardware aspects of timesharing. Phil Morse drummed up some money from, I think it was NSF, to work on timesharing and Herb Teager was going to develop a timesharing system. The problem was that Herb's vision was somewhat grandiose and he was going to do the whole system from scratch. He was going to write new programming languages. He was going to have to work out the terminals access to the computers. He decided that it was important that people have handwriting input to the computer. It was a giant menu of things that he was going to try and do. He also wanted to run a tight, little group and he just wasn't gaining any traction at all.

I saw this as being semi hopeless so I came up with the notion of couldn't we just do some sort of a demonstration system on the IBM 709 which would exhibit the interactivity so that people had some notion of what we were talking about. Because the amazing thing in those days was that you could talk yourself blue in the face to the people at IBM, to anybody who didn't program, who didn't understand what it meant to have only one shot a day at the computer. Not being able to debug on line was just incredibly hard to explain and so a demo seemed to be the order of the day. I started thinking about how to cobble it together and it turned out that by being artful we could sort of kludge the 7090- the 709, excuse me- into being a timesharing demonstration system. Initially we just called this an experimental timesharing system. We managed to get an interrupt clock. IBM had an off the shelf change they could put in, a so called RPQ [Request for Price Quotation], which they installed in our machine which gave us the ability to grab control of the computer. We were able to swap the computer contents in and out and we created a special version of the operating system which had 5K words set aside for a timesharing supervisor, [that] came to be called CTSS [Compatible Timesharing System].

Webber: What kind of skills did you need to be able to pull together this concept? Was it obviously experiencing the frustration of no online debugging but did you need personal knowledge of how to build hardware, how to connect remote pieces together?

Corbató: No. My navy technician background was invaluable so I had some idea of what was going on. The deal with IBM was of course that only they could touch the machine's hardware. Fortunately we had some liaison people who were awfully helpful. Loren Bullock in particular was the fellow who worked his heart out for us. He understood what timesharing was all about but technically he was just a salesperson, a relatively senior one at that but still he didn't have the clout inside IBM to make them do the right things.

Webber: You have a computer system, a prototype basically for what actually was CTSS. Just about to get into the real CTSS, the Compatible Timesharing System, and I think one of the questions that I have is what is compatible about it? What does that mean?

Corbató: The problem was that in those days in attempts to build interactive systems people were building specialized systems such as FORTRANSIT or the JOSS system or even the Dartmouth timesharing system. The problem was that you had to abandon everything you had ever done before. You couldn't use any of the older programs. So we were trying to create a workbench where with maybe a minor tinkering, [one could] take jobs that had been working perfectly in the older batch environment on the 704 or 709, run them without significant change in the timesharing system, work on them there and even go back if you wanted to, although people didn't need to. So that was where [we] were compatible and we were trying to emphasize that distinction. It was a very small point in retrospect but the thing that stuck was the acronym CTSS. People found that easy to say and no one bothered to say 'compatible' very often.

Webber: About when was this and when did it actually start getting used at the Computer Center?

Corbató: Well, let's see. We were first able to do a four terminal demonstration in roughly mid November of 1961. This work had been done with Marjorie Merwin, (who later [married and] became Marjorie Daggett), Bob Daley and myself. We [were] the three people that had done the hard work to get this thing to fly.

Webber: What were the terminals that you—

Corbató: Herb Teager had managed to cobble up four Flexowriters into being special terminals which could be interfaced to the IBM 709. For memories to store programs and files of these four terminals, we had commandeered four tape drives. Mind you, the machine only had maybe 12 or 16, I think. It was a very big, well-configured machine, but to take four of them and dedicate them to being storage space for these four terminals was obviously an extravagant experiment. As I said earlier, we had partitioned the memory so [that] we used only 5K words of the memory. Of the 32,000 word memory we used 5,000 for the supervisory program which included the buffering of the typewriter input, output of the four terminals. So it was a crude shoehorning in of the ideas which didn't totally shut down the machine for other purposes but allowed [timesharing] to be demonstrated. That was the initial demonstration configuration. By this time MIT had convinced, Phil Morse in particular, had convinced IBM that [they] really ought to upgrade the machine to a 7090 which [was] logically equivalent to the 709 but instead of vacuum tubes it had transistors. It was IBM's first transistor machine with a tremendous improvement in speed-- I don't remember what the ratio was-- and also reliability. Tube machines were really tough to keep going. So that was slated to occur in early spring of 1962. In fact I wrote the paper on CTSS called "An Experimental Timesharing System "in approximately January of 1962. I anticipated that we would have the machine swapped over to a 7090 by the time the paper was to be delivered in May of 1962 and to my chagrin the paper came out saying 7090 but in fact it was still a 709 in May. So I learned my lesson the hard way. Never, never say something that hasn't yet happened beforehand [in print]. I have since put disclaimers everywhere in sight on the paper or as part of the errata. It did become the 7090 by approximately August of that year but it wasn't May.

Webber: In August you had a 7090 in the Computer Center. Is that where it was?

Corbató: The 7090--

Webber: And it was used part time for the CTSS experimentation or—

Corbató: Yes, it was always part time. Because it was basically a batch machine, We could commandeer it now and then for brief periods without totally disrupting the flow of the work. Nobody was expecting real time response on their batch jobs.

So we got the machine converted to a 7090 in approximately August and we spent the next many months, maybe nine months, making some major transformations to the system. One of the key things we did [was] we got IBM's attention enough so that they allowed us to add on, with funding we got from NSF, an extra bank of memory, that is, an extra 32,000 words of memory. It was an unusual RPQ to the machine, Request for Price Quotation in IBM jargon, which allowed us to have the entire extra bank be the supervisor rather than trying to wedge it into the original batch bank. We also got a disc memory, an IBM 1301 disc, forget the number of words it had but maybe 9 million or so, which allowed us to use that for storing user files and for swapping programs in and out. So those [were] two key changes. Oh, and the third big change was [that] we got IBM to look around to see if we could get a terminal controller, a typewriter controller, which we could use to more gracefully get terminals into the machine rather than Herb Teager's home built four typewriter special hardware. That was the IBM 7750 which was a huge box which IBM had I think first built for the Sabre system, an airline reservation system they developed for

American Airlines. What that did was that it allowed up to 32 devices to interface to it. It did not include the telecommunications. That was another layer we had to add in.

We then began to work getting modems. The earliest modems were 110 baud modems which are incredibly slow by today's standards. That later got upgraded pretty fast. The modem itself was a box as big as a couple of shoe boxes, incredibly cumbersome. We got the help of Carlton Tucker, a professor of electrical engineering who was a specialist in telephone switching. He rigged it up so that we had telephone lines from the MIT dial up switchboard or dial up switch. We went from getting the 7090 in the summer of 1962 to the late spring of 1963 when we were beginning to run CTSS with this new configuration involving the extra bank of memory, the 1301 disc drive, and the 7750 hardware. We were in place to be the workhorse for the Project MAC summer study which was to take place that summer.

Webber: You said that you convinced IBM to do this or you convinced NSF to fund this or what not. It sounds to me like you had to do or someone had to do a lot of negotiating with the people that had the resources you needed. Was that a major part of your job during these years?

Corbató: Well, I was a participant. I was one of the arguers. There was a whole other theme going on of people agitating for a timeshared computer [for MIT] and John McCarthy was a key advocate. There were others too. I'm losing track of the chronology here.   I would almost have to retreat to some of the documents to pin it all down.

Webber: That's fine, but there was significant negotiating that had to be done and I was just curious as to how eager these other resource providers were to share or donate or contribute to the efforts. Apparently, a few years later there was a clear understanding that this was all valuable but it was probably groundbreaking when you were trying to do this or the team that you were a part of.

Corbató: Well, McCarthy's articulation of the goals and the vision was very seminal in my view and it got a lot of people's attention. IBM was trying to humor us. They viewed it as kind of an interesting experiment. Unbeknownst to us in this whole time frame, they were hard at work developing what came to be known as the 360 line of machines and they had it close to their vests. The people that had sold the management of IBM on the 360 had convinced them that it was a machine to solve all problems from here to forever after. That wasn't true. Clearly, timesharing was not on their agenda and they viewed it as an aberration. They saw the goal as to have a line of machines which allowed them to sell much like the auto manufacturers. You would have a starter machine and then a larger machine and then a larger machine.   They saw it only as a way to sell machines to customers. They did not think of it in terms of the way people would use computers. They saw it as a way to sell computers.   We're getting a little ahead of ourselves here.   When they eventually unveiled the 360 to us after we had signed nondisclosure [agreements], (we actually knew a few months in advance of the public), it was too late. They were just out of step with this whole direction of timesharing but that's another story.

Webber: Let's get back to the summer of 1963. You had mentioned Project MAC summer program. This is the first time you've mentioned Project MAC so why don't you go into that a bit?

Corbató: I realized that as I said it. We obviously left off a whole chunk of the story. A slight digression. I'm looking at an archival CD that I created in April of 1999 which tried to capture some of this early history. What I started to say was that there was a program called the MIT Science Reporter which was put on the public television station, WGBH. They actually wanted to tape in May of 1963 so this is just before the summer session. They wanted to do a demo and create a half hour program about timesharing. This was done using Ampex tape drives, 3 inch wide tapes, used for early television. They had a truck outside where they had the tape drives. They spent a whole day interviewing me and showing off CTSS in this program that was taped on May 9th, 1963. I'm having to look at notes here. Thank God I wrote it all down. It went on the air on May 16th, 1963.

And the reason there is a CD at all is that in those days I was asked casually as the program was about to air, would I like an outfit down in New York to photograph the program as it aired. In New York they proceeded to run the program on what was a called a kinescope, a big TV tube in effect, and photograph it with a 16 millimeter camera. I think I paid the extravagant sum of $75 or something to get this souvenir. And so I had this 16 millimeter tape sitting around. It's still in my den nearby. Somewhere in the 1980s I suddenly realized that no one could ever see this tape because it was hard to see in the 16 millimeter format. Nobody is really ready to see that. So I had it transcribed to VHS tape and then later when MIT was about to build this building, called the Stata Center. The late Mike Dertouzos, came up with the idea that something ought to go in the cornerstone, some souvenirs of the early days of Project MAC, and so would I come up with what I wanted to put in. Well, I put in some of the early papers but I also decided I would try to put in a CD. Through a laborious process which I won't bother to detail right now, I managed to transfer the VHS tape to a QuickTime file which resides in the CD. That's now in the cornerstone, so I'm told, although I've not been able to find out where the cornerstone is and I'm skeptical that it really is. Meanwhile, I have given out copies to anyone who wanted this snapshot of the early days including the papers and so forth. Anyway, this movie of my demonstrating CTSS in May of 1963 just before the summer study is in the snapshot. But I've taken a digression away from how Project MAC started.

Webber: What's the actual time frame on Project MAC? When did that actually start?

Corbató McCarthy was a key figure. But let me see if I can get all this right. Licklider, who had once been a professor at MIT, was out at Lincoln Laboratories and he got his hands on some computers there. He was fascinated [with the programming experience] and ended up going to BB&N, Bolt, Beranek and Newman, where he wanted to put a timesharing system up. There were a lot of [ongoing] efforts to create timesharing systems [then] including the Dartmouth timesharing system. [On] the BBN system there was a four person team.   McCarthy was one of them, primarily as an adviser, and Ed Fredkin and Shelly Boilen, along with Licklider. They tried to create a simple timesharing system based on the PDP-1 which today we'd call a personal computer, but it [was] still a relatively big box. It was the first of the mini computers. They got that system going approximately in the summer of 1962 because for a long time they were bragging that they were first until I had to point out to them that we had demonstrated ours in 1961 but [these are] small details. So McCarthy was promoting [timesharing] along with Licklider at BBN. Licklider then got recruited to go down to ARPA to embark on a program encouraging man, machine interaction. The story I heard was that the Cuban missile crisis in 1961 had shocked the Pentagon because they discovered that a lot of their information systems were practically unusable. They couldn't get information fast enough and they were crashing. It was just a mess. They were so shocked that they had such poor man-machine interaction with their computers, their communication systems and their databases and the like that they wanted to start research trying to improve on this and get better tools for the military's purposes. So that's how Lick got recruited down to Washington. Lick began to travel around trying to encourage people to embark on timesharing and he came up to MIT to see if anybody was interested.

Webber: It was after you had already demonstrated CTSS?

Corbató: I'm not sure. We were certainly about to demonstrate but the key thing is that Bob Fano, myself, Doug Ross and about a half dozen or so of us who were participants in the computing activities at MIT were at this meeting with Licklider. Licklider started to ask us what we wanted- what we would think about doing. But it was like a cat fight. Everybody had their own notion or plan or scheme. Licklider took it in stride but we were dismayed by our behavior in front of this potential sponsor who wanted to give us money to do something. We were so disorganized. It was disgraceful. Fano came back from that [meeting] and recognized that there was a lack of leadership at MIT. He began to think about it and subsequently had a long train ride with Lick coming back from a meeting down in Hot Springs [Virginia]. I think Fano would claim that he was persuaded to go ahead when he came back from that meeting. He decided that he would start to organize a venture which came to be called Project MAC. That formative stage was in the fall of 1962. Bob made the decision that we would use CTSS as a prototype for Project MAC to exhibit timesharing and we would do that by getting a carbon copy of the machine at the Computation Center. I think they placed the order just around the turn of the year, 1962 to 1963. The plan was to have a summer study with people invited from all over the country for a week or two at a time to come to MIT to spend learning about this new project organized around man-machine interaction.

Webber: The second machine, the Project MAC copy of CTSS, was that funded by DARPA--

Corbató: It was funded by ARPA, DARPA. They kept switching the name back and forth but it's the same thing no matter which you hear and yes, that machine was funded entirely by ARPA. It didn't actually come in until approximately October or November of 1963 because they couldn't get the machine built fast enough. So for the summer we actually borrowed the time off the Computation Center machine because the batch users couldn't tell we were siphoning it off a little bit, for the real time activities.

Webber: At the end of 1963 you now have two CTSS machines that are up and working, one at the computer center that's being used by people doing research in their work and then the one with Project MAC. Was that used entirely for research or was that also used for people that had just batch like jobs to do? How was that used?

Corbató: The Project MAC version of CTSS was used exclusively for timesharing. It was never used for batch and if people had a batch job they just ran it as a long timesharing job is my recollection. I don't think we ever used it the other way. One piece of jargon which you probably remember, Steve, was that the panels on the Project MAC machine were red and the panels on the Computation Center machine were blue so people tended to refer to them as the blue machine or the red machine. And the interesting thing was that we could dial up our computer terminals to either machine depending on the number you picked because we were using a telephone switch and had a 7750 on both machines.

Webber: You have many users now using it in the 1964 time frame. I'm assuming—

Corbató: 1964?

Webber: At the end of 1963 when you finally got it, then you're actually running the Project Mac machine as a timesharing machine. What were some of the obvious early improvements that you could see that should be made in that? I want to get on to Multics that really demonstrated a great deal of advancement.

Corbató: Well, first let's back up a little bit.   We'd just gotten going and were catching our breath in a sense. Project MAC was in a different building as you know. 545 Technology Square is what it came to be known as. The summer study was held in that new building. Why was it held there? Because there was no space on campus for a group of people as large as Project MAC. Initially it was two floors of the building and eventually came to be all of this nine story building. So that was a long history. We thought of CTSS as a prototype, as something that would exhibit the idea, demonstrate it and indeed it did. We enhanced it quite a bit in the process for about a year or so when we improved the file system and I think we switched to a higher performance disc drive, a 1302 disc drive. There were some [other] enhancements too. We went from a one level file system to a hierarchical file system on CTSS. But the key thing we did was we--when I say 'we' I mean all of Project MAC-- began to think about a new machine. The new machine was undesignated. IBM had kind of assumed that because we were using their equipment that it would probably be their brand of equipment. They in turn had the 360 still under wraps but which they assumed would be an ideal candidate once they got a chance to show it to us. And we in turn started drafting up some ideas of what we thought would be good and began scouting around, talking to different vendors and the like. That's where things got kind of dicey.

Webber: You said that you started to go beyond a prototype concept for CTSS. You really had a working system. I actually used it for years and it was much more than a prototype, it was an actual very functional system, but there were some limitations as you said. You could see much broader vision. You started to mention that it was dicey to come up with a hardware platform. Obviously, that's going to have a big play. MIT is becoming at this point well known for their computer science amongst just a few universities out there. Can you go a little bit more into what kind of specifications you were coming up with and why you didn't think certain machines would work and others would? It obviously had to do with what you thought the timesharing system was going to need.

Corbató: We began to write some position papers and I don't have my hands on them right now but spelling out the kind of things we expected to see. I would say the requirements got codified eventually in the six papers that were presented at the Spring Joint Computer Conference of 1965. There we spelled out what we expected Multics to be. Of course, we didn't have the name 'Multics' picked out yet [when we were choosing a vendor]. I don't think we did, I'm trying to [recreate] the chronology. What we did do is we started a group. It included Ted Glaser who joined Project MAC because of his interest [in the project] and was a brilliant blind programmer engineer who came to be one of the key leaders of the development of Multics. Also myself, Bob Graham, Jack Dennis and Doug Ross were involved. We started visiting vendors to see if they would be interested in building a machine with a new form of architecture to support timesharing. We went to CDC which was under the thumb of Seymour Cray. He was interested in building super fast machines and didn't see the need for changing his mind about anything.

Corbató: McCarthy was part of this group too [that went out to see CDC]. This was pre Project MAC, I think, but it doesn't matter. [There were two rounds of visits to vendors. The first was pre Project Mac and part of the Long Range Study Committee a few years earlier. This was when the McCarthy at CDC anecdote occurred. The second round of visits was after Project Mac formed and included the visit to Burroughs. FJC] McCarthy was kind of a wild man and he liked to make provocative statements. Seymour Cray snuck in the back of the room [during McCarthy's presentation], I heard later, heard about 20 minutes of it and walked out. It just wasn't what he was interested in, but that's a small aside. So in the process of trying to scout out a new machine, we visited a lot of vendors. We went down to Burroughs. Actually, it was at Burroughs that Ted Glaser heard about our interests and that's when he got interested in joining our project. Joe Weizenbaum suggested we might look at GE equipment. Indeed, we did look at GE equipment and of all the vendors we looked at, they were the ones that seemed the most eager to please and hungry. They had an architecture for the equipment which was more appropriate to being modified for a decent timesharing system. And so the combination of things led us to begin to work closely with them. Their top hardware designer at the time was John Couleur who was very amenable to change, too amenable probably, and so he and Ted Glaser proceeded to tinker with the hardware to the point where we had lots of ideas tangled in, segmentation and paging, things that were not then standard in computers. Segmentation isn't even yet.

Webber: You had mentioned that you went to Burroughs and CDC but IBM I assume was early in on that survey as to companies that might work but the 360 didn't look appropriate or they weren't interested?

Corbató: It looked very inappropriate. It was designed for batch processing and didn't have even a decent interrupt structure. It was just not the machine. It certainly didn't allow for multiple processors. It didn't allow for multiple memory banks. It didn't have the notion of an expandable capacity for dynamic reconfiguration. They were notions that were just alien to what they were doing. Their notion of reconfiguration was to ship you a new machine which was fine for the sales department but not for us.

Webber: Back to Ted Glaser and who else did you say was working on the hardware? John Couleur?

Corbató: John Couleur at GE. It was then GE. Later they got bought up by Honeywell. Well, the GE computer division [to be precise]. Now one of the consequences of that was, as everyone knows, we elected to work with GE to build Multics. Also Bell Labs was interested in acquiring a new computer system. They were quite intrigued and sympathetic to the notions of what we were doing. Ed David down there was a key figure and an old friend of Fano. They decided to become partners and the interesting thing is that the three organizations agreed to work together. But nobody was in charge. A critical flaw in retrospect because it made everything have to be by consensus and meant the ability to reject ideas or to make hard decisions much more difficult. One of the side effects of our electing to work with GE was that IBM panicked. They got very angry. They were ruthless. Their standard mode of operation was to punish the salesman who'd failed. It was really a tough story that I've only heard about in bits and pieces. The management didn't understand because they had been misled. One of the key reasons of course is that IBM, like most large companies was run by people who had never programmed in their lives so they had no idea what we were talking about. They were relying on the expert advice from their engineers and they had been led down the primrose path. So they panicked. They thought because GE was a big company that they would be a serious threat to them because they had deep pockets. What they didn't understand is that GE wasn't organized as a computer company. GE was organized as basically a company that managed shopkeepers. Each of their divisions ran a little shop and was a cost center. The computer division was no exception. There was no way that they could take on IBM's depth and muscle. So that was a miscalculation but it led to them immediately trying to commission the IBM 360/67 which they tried to build in forced draft mode. We had at our peak 100 programmers and they had at their peak 1,000 programmers.

That is my recollection of it all. They got it out a little bit before we did. We finished Multics but it took us almost four years and IBM got theirs done in maybe 2-1/2, 3 years, but it didn't run worth a darnn. It was lame as a duck because you can't force quality programming out the door. I think they had something like a 17 pass assembler or something because they tried to rush it. So they ended up floundering. Another sidebar is where they saved their bacon by having another system available. Some people that had worked on CTSS had built a Virtual Machine system, the VM system, which was a timesharing system, and which basically allowed them to save face and keep their customers happy. So they came out of it okay but there was a lot of anguish. Well, we were back tracing down the path of going to a lot of vendors and I guess we did.

Webber: Right. You've now selected GE, Bell Labs has joined the team, you're starting to staff up, hire programmers, designers and what not. This must be late 1965, early 1966.

Corbató: Yes. That sounds roughly right.

Webber: The six papers I think came in the fall joint computer conference of—

Corbató: Was it fall or spring? I don't remember. [Note: It was Fall 1965. FJC]

Webber: Anyway, in 1965 there were a lot of very novel ideas that were put into the press at that time. I'm wondering how those crystallized enough so that they all were compatible with each other and they made sense and I'm assuming it was all from the experience of CTSS but I'm sure there were other ideas as well. Can you say a little bit about those six papers because those are—

Corbató: Okay. Good point. The reason there were papers at all, because that was not the normal mode for most system programmers of the day, even today it is not the normal mode, was that Bob Fano recognized that if we didn't specify our goals, we would never get any attention and that we'd better basically articulate what we were trying to do or we would just be lost. So he encouraged us to get it all written down and to spell it out in advance. Trying to articulate the plan in advance was novel then and novel now but it was tremendously important in retrospect because it unified the vision of what everyone was trying to do. It set a goal, a set of goals, and it was tremendously important in recruiting people to come work on it. To this day the people who worked on Multics are singular in the sense that they think they worked on something very important because they had a vision of what they were trying to do and they saw that they had some part in making that happen.

The key goal of Project MAC was to influence the industry. Fano was a key leader in this [attempt] to influence the industry to change its ways, to change the way we built and used computers. Well, I would say in retrospect we partly succeeded. We certainly got people to pay attention. We didn't get them to change their ways so quickly, but what was introduced was the notion of interactive computing. Nobody doubts that that's the way to use a computer today. It was helped by the fact that the mini computer evolution came along. At first mini computers became affordable [to small organizations] and then the personal computers. Perhaps this whole evolution was predictable but nobody was predicting it exactly the way it happened.

Webber: I actually remember sitting down with you and Jerry Saltzer once and I forget which of you said it but you had said that the entire Multics system will fit in your briefcase and I have been through these acres of rooms of Multics computers and I knew that was obviously not true and never would yet it not only fits in your briefcase, it fits in your pocket now, the advances that have been made in the industry, so definitely there's been some progress there but from my view, and I actually am one of those people who is proud to have worked on it, there were many innovations that came, first in Multics. I often call Multics the ancestor of all of today's operating systems because many innovations came there. Who were some of the key people that had the ideas, the dynamic linking, the symmetric multiprocessing, the hierarchical file system, the security, access control? This just goes on and on. Everything that we accept today appeared first in Multics. Who were some of the key people there?

Corbató: Um...

Webber: There were only 1,000 or so.

Corbató: That's quite a list. Let's start one by one maybe. So which one do you want to pick first?

Webber: Dynamic linking. That's—

Corbató: Okay. Dynamic linking was something that I had a hand in back at the Computation Center. The seeds of that were set back early. We were trying to get a scheme whereby you didn't have to preload a huge complex of programs in order to interact. And how would you do that? Well, the obvious way was to follow a thread of load on demand. That fit in naturally with the notions of paging which had been pioneered in England with the Atlas computer and these all seemed like good ideas. Jack Dennis was the one who argued successfully from our point of view that the notion of segmentation was important and valuable, that the programs weren't all one big wad that was all welded together.

Webber: Was Jack Dennis the one that came up with the idea of mapping files directly into memory and making them directly addressable?

Corbató: I don't believe he was. I think that got hammered out in some of our early design sessions where we had people like Vyssotsky, Glaser, myself, Bob Graham and Peter Neumann. I think Vyssotsky was the one who was more emphatic to argue that it should be one-to-one so that a file and a segment were the same. So we adopted that as a realistic design at some point.

Webber: There is just an endless list of lessons that we learned. I know one lesson I learned was the value of simplicity. In the early Multics design, there was a very complex paging system with different page sizes and all kinds of stuff and the first boot load took an hour and the key there was making it simple because it really made a difference. That was a lesson learned. I'm not sure how quickly it took to adopt that—

Corbató: Well, necessity is the mother of invention as you know. We all learned. That was a brutal period when we discovered that we had built a monster. We had several instances of that inside Multics. The initial paging and booting [for example]. Yes, one of the problems was that we were all kind of young and naïve. We couldn't see that we had built something that was too complicated. Only under the duress of realizing it was awful did we dig in and realize it didn't have to be that way. We couldn't breadboard the system. Today you would try much harder to throw something together, get a skeleton working, and then try to embellish it rather than build it first and then try to streamline it. So that was a hard lesson we learned throughout the project. And it wasn't just there. There was a device called the GIOC, the Generalized Input Output Controller, which was a very novel, too novel input output machine which Ted Glaser had cooked up largely by himself. [The initial sofware] was a monster too and again went through a similar cycle of [an] initial design, [which] worked in principle but did not work in practice. It was complicated too by the three organizational mix where we couldn't reject things easily, not that we were that good at spotting what should be rejected. When they went to build UNIX they started just the opposite way and with good reason because they were a small team. They did not add in too much because they couldn't handle too much. So they started with a skeleton system and built up and that was the right way to go.

Webber: In a sense, you started with CTSS and then built up but the first big step was so gigantic. What about the various unique hardware things. Not only were there multiple processors symmetrically configured and multiple memory controllers but they were dynamically configurable so that you could add a processor and take it away. Where did all those ideas come from?

Corbató: Well, that was part of the vision of a computer utility. We really believed that one of our things to demonstrate was the notion of a machine which was expandable on the spot and contractible. In fact we used that in the later stages of the project where we would run a partitioned system to test out new versions of the system without harming real users. So that was of practical value but that hasn't caught on for obvious- well, maybe not obvious reasons. I am not involved directly with big server farms and things like RAID Arrays and whether or not they are dynamically modifiable but I suspect they might be. Tandem Computer when they were in their heyday I think had some dynamic reconfiguration.

Webber: Right. I think one of the challenges that the Multics project had was that the dynamic reconfiguration was of shared resources whereas the large server farms now, they're pretty partitioned and they take advantage of networking and meshing and what not so I think it was a much harder problem than Multics actually solved. Some of the other key things in Multics: One of the first things that came out was access control lists so you could control what you could do with a file and in fact who could do it. I assume that Multics was the first place that occurred. That's the first place I've heard of it. Whose ideas were those?

Corbató: Didn't we have some of that patched into CTSS a little bit? The notion of shared files on CTSS was that there was a shareable [file] place and you had your own private [file place]. It was a rock hard partitioning but it seemed like a logical extension to try to make the notion of access control be part of the machinery.

Webber: Another example of the complexity was that Multics generalized the concept of supervisor state and user state, to have rings and when a file had to be in multiple rings at the same time plus multiple access control lists it got so complex, not to implement it. We actually implemented it but it was so hard to explain to people that we'd just cut it back and said no, we're going to make it much simpler.

Corbató: Right. That was a lesson we learned a lot over and over again. In fact [that is true] even today in UNIX systems which adopted a large part of the machinery that Multics has for access control. I think getting access control right is still an art.

Webber: There have been many advances in I guess it's called security management now or security in general. There's a lot of research that's gone into that.

Corbató: Well, that was- but I think you just said the key word. Security was still- was an issue then and it still is today and remember in the early first computer systems there was no protection at all. Anybody could write in anybody and- or read in anybody and even the IBM 360 you could read somebody else's files even if you were coexisting in the machine and it had nothing to do with that so there was some real naiveté on their part in designing that. They didn't see the need to protect it. The assumption that there is this single user using a resource was slow and disappearing and where today we take it for granted that everybody and his brother is on the same machine.

Webber: Right. As a matter of fact, Multics was the first machine to be certified B2 compliant with the multilevel security thing from the government. I think we called it access isolation mechanism. It let secret and unclassified people use the same machine which was very difficult on a shared resource. That was another thing that first occurred in Multics. The list of Multics firsts is very long. One thing that I remember from Multics is something that we in the back rooms called the Multics way and basically that was if it weren't done the right way with the right review and the right standards and the right programming conventions you were laughed out of the room and so if it really weren't done the correct way it was embarrassing to you. Who instilled that in the whole team to make that the case? I know Bob Daley had a part of it but it was a probably a shared thing. What would be your view on that?

Corbató: Well, I never heard that expression of it myself. I guess I was part of the problem—

Webber: Trust me. It was there. There was the Multics way.

Corbató: Well, we had some fairly good standards. First of all, we had this notion that if you couldn't say what you planned, you couldn't [do it.] In the early days of programming, people just sat down with a blank piece of paper and started coding. They wouldn't show it to anybody else until they were finished and so forth. We had a different view of it. We had more of an engineering approach where you had to say what you planned to do [and] you had to run it through some sort of design review. Then the novel part for us was [that] we also said that you had to build it yourself. In other words, in order to get the right to build something, you had to be able to say what it was going to be. That was relatively novel and disciplined [compared to] the programming that was often done in other places. But we had to do it that way or we would have been out of control. We didn't have the luxury of passing it off to some team of programmers to go do it and that probably made it a lot better because of that.

Webber: I remember Noel Morris pretty much reviewing all of the code that was written for a few years, auditing it before it could be submitted into the normal library of accepted code and he was one of the guardians of the Multics way, if you didn't do things the right way you would bear the wrath of Noel.

Corbató: I didn't know that but I believe it. I know Stan Dunten was involved in some of those [code reviews]. In the instances when two people read through the same code, one wrote it and one read it, it was invaluable because it would shake out so many lame spots in the code ahead of time. You didn't have to debug it, you could just spot [the bugs]. In retrospect I would say that is probably a very good design discipline even today. No matter what you're building, if you had a second person forced to go through the logic mentally beforehand, before you tried to implement it at all, you'd probably shake out most of the problems.

Webber: Every year you would go down to Washington and get us more money or at least that's the way I remember it. You would get another round of financing, Multics lives for another year. Can you say a little bit about that?

Corbató: Well, it wasn't me so much. I was part of the support team but the people who were really saving our bacon were people like Bob Fano. Ted Glaser and I would also go down and sometimes [Bob] Graham. The key thing that was going on there was that the sponsors had to be resold. Now one of the difficulties we ran into, and this is one of the things in retrospect, that is mentioned in a few of the papers, is that the hazard of a long project is that you get turnover. You get turnover inside the team that's building it and you get turnover inside the sponsors. And, lo and behold, GE got bought up by Honeywell. I don't know if there were changes at Bell so much but we had changes even in Project MAC where Fano decided his job was over and he could step down as director. Licklider who had been our patron down in Washington seemed like a wonderful choice as [the new] director. But it didn't work out that way at all because Licklider when he'd been patron had been very generous but when he was the director having to answer to somebody else, [he] didn't always understand what we were doing. He didn't have the mark of a traditional manager by a long shot and in fact Licklider was almost an anti manager. So we had some real reselling jobs to do and because it dragged on for almost four years, people were beginning to falter. In fact Bell Labs eventually pulled out because they had represented their system [internally] as being something that was going to be of great, immediate utility to them to be used within a year or two.

Webber: Speaking of utility, we never even mentioned what Multics stands for and I think the last word is key there. What I was told was that it wanted to be as reliable as dial tone on the telephone and that's very difficult of course. What does the acronym actually stand for?

Corbató: Multiplexed Information and Computing System I think.

Webber: Service.

Corbató: Service.

Webber: That was the key thing. We had to provide a service and that's part of the strength of it is that if you realize you have to provide something that's always there as a service is like the electric light switch or the telephone. It means you have to look at things totally differently.

Corbató: Well, I think actually you touched on a very good point there which reminds me that one of the key contributions we made in Multics was, (and I take a lot of credit for this), when we decided we were going to put programs on disc, I recognized fairly early in the game, as a programmer myself, that losing your programs because of a disc crash was a disaster and you couldn't accept that. Backup schemes were from day one part of the framework and we had a very highly organized backup scheme where Multics had demons running in the background and could detect when files had been modified. On basically a half hour cycle as I recall we would try to dump those onto mag tapes and have backup tapes up the wazoo. So indeed people did accept the fact they could leave their material on line safely even though we did have hardware crashes. Hardware was so flaky in those days. People don't remember and don't realize how flaky things were. We used to crash all the time.

Webber: There were actually some software crashes too.

Corbató: Yes. Well, actually the problem was you didn't even know sometimes what the cause was, you just had crashed. You used to have to program and think so defensively. Today's youth who pick up a personal computer and put their life history in it and don't even think about it, they don't understand.

Webber: : That backup software was one of the first things I worked on and I worked on a very early version and it was called Embalmers because it was going to preserve it forever, whatever you had written. We changed that name—

Corbató: That had a ghoulish overtone.

Webber: What do you think some of the important legacies that came out of Multics are? For example, what are you most proud of? There's the whole project of course but—

Corbató: Well, there are a lot of things. That's a good question. I hadn't thought about it directly. I think one of the things I'm proud of is having had an influence on the computing world and that influence has been in a lot of ways. One of them is through the papers which are archival and which I'm sure still get read. Probably the most important thing that we didn't recognize and plan so much, is [that] it was a training bed for an incredible number of professionals who went out to play roles all over the map, yourself included, in advancing the computer world. In thinking about [it] I feel proud of the fact that Ken Thompson came in, learned, got a tremendous exposure to all the ideas, participated in some of them and when Bell got jerked out of Multics, went on to do such a constructive thing as to build what's become UNIX. It's taken on a life of its own and still persists in every way, shape and form from Linux to Apple's MacIntosh OS X. And so it's permeated so much of the field and it's permeated also the people. That's a good feeling. I don't think we've had any obviously bad ideas that should have been killed.

Webber: I think it's true that the people that you kind of seeded the entire industry with by being trained with Project MAC and on Multics, there are many people out there that have claimed to have worked on it that didn't because I was there for many of those years and so it's obviously quite a thing that people are proud of.

Corbató: Well, one of the things that we should take note of, because both you and I know about it, is Tom Van Vleck's work of love which is to build the multicians.org web site which is an encyclopedia of Multics and all of the lore and fond nostalgia.

Webber: Is there anything else you would like to say? Do we want to wrap up or-- We have another 40 years of your life. Do you want to continue it now or—

Corbató: Well, I—

Webber: We have to finish this tape and so we can stop that now. Do you want to say anything before this tape ends?

Corbató: No, I think this is at least a stopping point. It's been a bit of a jumble but I think if people really want to dig further they have to go to the bibliographies and start reading some of the papers because it's written down all over the place but it's more tedious to read it of course.

Webber: Thank you very much for this great interview.

Corbató: Thank you.

END OF INTERVIEW