Interviewee: Douglas Comer

Interviewer: Fan Yuanyuan from Oral History of the Internet project

Date: March 23rd, 2023

OHI: OK, so today is March 23rd. We are honored to meet Professor Comer through Internet, which we can't live without a single day to do the oral history interview. So. I will just briefly introduce the project, so you can have a rough idea of what we have already done. So, the project was launched in 2007, recording and preserving the personal narratives of global Internet pioneers and the key persons about their contributions to the development of the Internet. By the 50th anniversary of the Internet, which we believe is 2019, we have already interviewed more than 400 key persons around the world. So, we are trying to interview 500 persons in our first phase. So, we'd like you to share some of your personal experiences and stories with us, especially those you can still remember in details. So, it's just like talking to a new friend who is keen to know more about you. So, would you please start from your name and your birth date?

DC: Douglas Comer. I was born on September 9th, 1949.

OHI: So, that's the true right answer. I've searched it on the Internet. There is so little information about you. I thought 9.9 might be a random date on the Internet.

DC: Well, I haven't participated in many social media things. Everybody who needs to find me seems to be able to find me just fine.

OHI: Yes, you're right. So, where were you born and what's it like to live there?

DC: I was born in Vineland, New Jersey. We actually lived in a very rural community outside of the town. The town wasn't gigantic, but we lived way out in what people called the sticks. When we first moved to the house where I grew up, it was a dirt road and it was in the middle of farmer fields across the street from us was just a farmer's field. But, you know, times go on and now the road is paved. But when I was growing up, it was very rural.

My parents were not educated. They didn't go to college. No one in my family had gone to college. In fact, none of the friends had gone to college. So, for me, it was sort of isolated and lonely. I was interested in technology and I got people to give me old radios and televisions and I played with them to figure out how they worked.

School for me was wonderful. It had a big advantage. It was during the Cold War. And during the Cold War, science and engineering were important and everybody wanted to be in science and engineering. My high school had hired a physics teacher who was a retired physicist. And I thought it was one of the best courses I had ever had in high school. Here was a real expert. Unfortunately, the parents in the town were very upset that I was the only one who actually did very well. Most of the other kids didn't get high grades and the parents said, we need to fire this teacher.

In math, I had two teachers who were absolutely wonderful. They had PhDs. They had hired a PhD in math. Can you imagine in high school? But remember, this was the Cold War, so they were trying to emphasize mathematics. And I thought they were wonderful teachers too. And it was very interesting. You know, when you're a kid, you don't know anything about the real world. You don't know who you are and how the real world works. So, I had these wonderful teachers. One of them called in my parents and told my parents, your son needs to go to college. And had even gotten me into MIT, talked to people at MIT. The two professors were from MIT, believe it or not, a leading research institution. And they had recruited these people to be high school teachers. My parents were convinced I should go to college. And I went to college. I majored in math and physics. And that was the start of my career.

OHI: So, you are kind of like a Sheldon in real life. The character from Big Bang Theory.

DC: Well, I guess I was in a way. Yeah, I was very unusual. But again, when you're a child, you don't know that. I used to tell my friends in high school, sort of discussed that they would get low grades and I would say, you're just not listening. It's really easy if you just listen. You're not listening in class. I didn't realize that it was easy for me, but very hard for them.

OHI: So, you're a genius when you were a very little kid. So, when did you start to realize that you were different from others?

DC: My parents, evidently, when I was a very small child, when I was first starting school, when I got to first grade, they were giving these tests, IQ tests. And my parents were told that I was a genius. But my parents decided that it would be very bad for me to know that. So, instead of telling me I was a genius, they told me, you're just slightly below average.

So, I grew up thinking that things that were easy for me were super easy for others. And when you're a kid, I guess you know this when you're in school, you meet people who brag all the time. I'm really bright. I know everything. And I would always believe that. I would believe, oh, yeah, they know everything. I don't know much.

When I got to grad school. I was still convinced that I was just an average person. In fact, by the time I was a professor.

But I kept succeeding when I was in grad school, we had these very tough exams called PhD qualifiers for a PhD program. And I was the only person to pass it. So, all my friends took that all, you know, my colleagues in grad school took the exam. I was the only one who passed. And I was convinced that they just didn't even work at it.

One day after I was a full professor. My mother was visiting. And she said, well, you know, you are a genius. And I said, well, but you told me I was below average. And she explained that she didn't want me to be egotistical or conceited. And so she thought it was a good thing to do.

What can I say? I really believe that most people weren't working hard.

OHI: So, this kind of encouraged you to work harder because you believed that everyone else is much better.

DC: Yeah, it did. What can I say? The good news is that for me, a lot of things turned out to be really, really easy.

OHI: So, do you also like Sheldon have a sister and a brother?

DC: I have a sister. And I guess in a way like Sheldon, my sister is completely different than I am, not technical, not especially good at math and science.

OHI: But she's quite good at dealing with people?

DC: She has strengths that I don't have. Yes. She's more empathetic.

OHI: OK, so when did you start to realize that you are interested in engineering or machine something?

DC: When I was a kid, I was very interested in everything technical. Electricity, how does it work? How does a motor turn? You know, I wanted to know all that stuff. I read books. One of the best things was a free public library. I could take out books and read them. And I read books on electricity. I read books on electronics. I figured out how in those days it was things like vacuum tubes. But I wanted to know, how does a radio work? How does a telephone work? And I read a lot of books and I got a bunch of old stuff and I just experimented in my basement. You know, my parents would let me play with these things.

I have to tell you, in the beginning, I blew out a lot of circuits. I would hook them up, didn't quite understand voltage and current and how to do it. And I would end up in. And by the way, in those days, it was all vacuum tubes. So, there was high voltage. And, you know, it wasn't just five volts. There were very high voltages. But I eventually started to learn when transistors came along, I was super eager to get a transistor. I wanted to get a transistor. See what it could do. And they were expensive. My parents were not rich, they were very poor. So, I couldn't really buy new parts. I had to wait until I could get someone throwing something out, you know, an old radio, take things out of it. But I was very interested in all of that. I just didn't have any way to learn other than books.

OHI: So, you try to gather all of those parts from litter where someone threw something and you try to pick up something useful for you.

DC: Yes. My father brought home my father worked at a factory and they had some control things he brought home, for example. He would go to the junk pile and see if there was anything electronic and bring it home. He brought home a photo cell. It was it was supposed to be controlling an outdoor light when it got dark, it turned on the light. So, I had a photo cell. I played with that. So, yeah, it was all junk parts.

OHI: So, you play all you can get. And the final again together to be a working machine.

DC: I had cobbled together a television out of broken televisions. So, I finally got that to work. And then a guy was giving away an old color television. Now, this was in the days of black and white TVs. My parents couldn't afford a color television. That was a brand new, expensive thing. And this guy was throwing one out. And I asked if I could have it and he gave it to me. And I got it to work. And I had the first color television in our family.

OHI: So, you've saved a lot of money for family when you were a still kid.

DC: Well, yeah, we didn't have much money. So. You know, they say necessity is the mother of invention, that if you don't have enough money, you figure out how to do with less.

OHI: Yes. And there is a saying in Chinese that saving money is actually making money. So, you've already started to make money when you were a kid. So, in school, you are kind of knowing things and learning things much quicker than other kids. So, how is your relationship with them?

DC: Oh, I was a nerd. I didn't have. what most people think of as real friends. First of all, we lived out in the country. You know, there weren't any houses around. It wasn't as if we were in town with other kids nearby. So, there weren't any other kids my age when I was growing up. And I was very interested in technical things and most kids weren't. So, I didn't have many kids to share with. There was it was one kid in school who was also from a very poor family and he was cobbling together televisions and making them work. And we talked, but it was mostly about technical things.

OHI: So, it's kind of like you are much more mature than other kids.

DC: Yeah, I know, people used to always say that I was a little man. little meaning, I was still small, but. And I didn't understand at the time. Lots of people say it. And I didn't quite understand what they meant.

OHI: So, did you have regrets that you have lost your childhood because you've already grown up when you were a kid.

DC: No, I don't. For me. I was not interested in the same things as most kids are. My parents would go to the boardwalk and we didn't have enough money to go on the rides. All the other kids were clamoring. You know, they wanted to go do the rides, ride on a Ferris wheel, ride on a roller coaster. I wanted to know how it worked. How did they make the thing stay on the track? What was the design? So, I didn't have the urge to just play. I had the urge to figure it out. I know it sounds nerdy and what can I say? I just found that so interesting.

OHI: So, you were quite enjoying it. Although maybe to other ones, they are quite boring try to figure out how it works.

DC: Yes. I always wanted to know how things worked. And how to build them.

OHI: So, how did you get admitted in MIT?

DC: Well, the teacher in my high school was from MIT, talked to the professors and said, we should send this kid to MIT. My parents said, absolutely not. They were very religious. They were very worried that MIT was not a religious institution. Besides, you had to pay for it, I was going to get free tuition, but I had to pay for a place to live in Boston and Cambridge, and it was very expensive. So, they said, absolutely not. If you're going to go to college, you have to go to a religious college. So, I went to a so-called church school. Their church had some schools affiliated with it, and I went to the best one of the schools that they allowed me to go to.

OHI: So, you went there and study math as your major.

DC: Math and physics. I had a double major. Remember, I told you in high school, they were the best teachers I had. They were very knowledgeable. So, I started math and physics.

OHI: So, have you ever imagined that if you went to MIT, everything will be different?

DC: Yeah, I suppose it would have been. One of the things that I've observed about students, and I always tell them this is, if you go to the best place you can get in, you will meet people who stimulate you. If you go to the worst place you can get in, you'll be the best person there, but you'll be lazy. You won't learn as much. You won't be forced to work hard. So, I suspect if I had gone to a different school, I would have been different, but you never know.

OHI: But you still achieve a lot of things that lots of people can’t achieve in many, many lives.

DC: I know. It's been fantastic. Here's the thing. You can always start where you are and figure out how to do more. I know a lot of students that come to college think, well, I'm here in college, and I'm going to learn everything I need to learn, and then I'll go out and I'll be famous. And it doesn't really work that way. Everybody has to figure out who they are, what they're capable of doing, what they're capable of doing that's hard for other people to do. And then they have to capitalize on it. So, it's that matter of figuring out the world and who you are.

OHI: That's a question that lots of people can't figure out, even when they were quite old.

DC: I know.

OHI: It's a difficult question. Not everyone has the luck to figure it out.

DC: But everyone has talents, capabilities, and interests. I have met some very interesting people in my life. I met a fellow who was a gardener. He did landscape gardening. He had a truck, he had a rake, he had a crew, and he would go around and mow grass and clean up landscaping and put in trees and take them out. And he absolutely loved it. He was good at it. He loved it. I couldn't imagine doing that. Can you imagine every day you have to go out and dig holes and plant things and you wouldn't get to do anything technical? But everybody has things that they do well. And they have to figure out what it is that they do well.

I just saw an interesting family. We were on vacation and we saw this family. And there was a brother and a sister. And the sister was struggling. They were playing mini golf, a little game. And the sister was very, very tiny and she was struggling and messed up several times and started to cry. And the brother, not the mother or the father, the brother rushed over, held her, reassured her, told her everything's going to be all right. That kid has incredible ability to care for people. I hope he realizes when he grows up that that's a real talent. He could make her smile and reassure her. It was amazing. So, everyone has some talent. It may not be in computer science or in networking or whatever. It may be in caring for people. You have to figure out who you are and what you can do.

OHI: And doing what you loving. And it's lucky that you can do what you loving as a career.

DC: Yes, I have had the luxury of doing exactly what I wanted. It's been great. When I was a kid, I decided everybody ought to figure out what they want to do and what they're good at. And they take the intersection and they pick a job. You know, there's so many jobs. You can find something that you want to do that you really love. I did it. I wish everybody could do that. I have met people who did that. I told you the gardener, he just loved getting up every day. He would tell me, "I love this job. I meet people. I make their yards beautiful." He loved it. And for him, that was not hard work. I have put in tremendous number of hours working. But for me, it's never been hard work. It's been enjoyable. I often can't wait to get the day started and do more.

OHI: I'm kind of envious of you for having that energy and power to do all those you are liking. And so, when did you start your study at Houghton College?

DC: Yeah, 1967, I went to Houghton College, graduated in 71. I decided to go to grad school. My math teachers told me, "you should go in math." My physics teachers told me, "no, no, no, you should go in physics." I said, I'm going to go in computer science. And they said, that's not real science. The mathematician said, that's not real math. The physicist said, that's not real science. You're wasting your life.

OHI: You're wasting your talent.

DC: But I had gotten a computer, I had the luxury of playing with a computer. So, the college got a computer my junior year, and it was just amazing. I got to play with a computer. Of course, there was no one around to teach computers. They had gotten it to use in their administration to process accounts receivable and payroll. But they let me play with it. I taught myself how to program. It was just complete and utter fun. And I decided to go to grad school in computer science.

OHI: So, under what kind of circumstances did you get a chance to play with a computer? And when did you start to realize there is a computer in your school?

DC: Well, we knew when they were getting it. It was a big discussion. They were getting this computer. They needed somebody to operate it. Now, those days it was punch cards. So, the data was on punch cards, and they needed an operator, a person, to take these punch cards, start a program, put the punch cards in the computer. You had to put in a few at a time. You had to keep putting them in. And then when they go through the computer, you had to keep taking them out to keep the computer running. You had to put in the paper, you know. So, they asked, would you like to work in the computing center doing this job? It was a manual job. It wasn't programming the computer. It was literally a manual job, taking cards, putting them in, wait for them to run through, taking them out, putting them back in the drawer. And I got that job. And I said, what about at night when the computer isn't being used? And they said, well, if you want to play with it at night, here's a key. You can play with it at night. It was in the days when they leased computers. You leased them. You rented a computer, and it didn't matter that you were using it or not using it. You paid the same rent. So, at night, after I finished my classes, I would rush to the computing center, figure out how to use a computer, how to write computer programs.

OHI: So, you had the chance to play with computer for four years?

DC: For two years. And so, let me tell you what happened. They had a program that required 12 hours to run. 12 hours. You have to go back to the 1960s and 70s, early 70s. Computers were very slow, and they used punch cards, and were very, very slow. So, it took 12 hours of pushing cards through this computer, and it was the grade program. It was the grades at the end of the semester. 12 hours. And I kept asking myself, can I do it faster? Can I write a computer program to do it faster? Now, the computer was extremely small. 16,000 bytes of memory. 512,000 bytes of storage. That was it. 512,000. And, of course, the operating system, the programs all had to be on there. So, you didn't have much space at all. How can I make this program go fast? I thought about it, thought about it, thought about it, and thought about it. I just got that in my head. I couldn't stop thinking about that problem. I went home over Christmas break, and I had been thinking about this problem day and night. One day, I woke up after sleeping all night, and I had the solution in my head. I grabbed a paper, pencil, and I wrote it down as fast as I could possibly write. I know how to solve this problem.

So, I did. I wrote the program. It worked. And I showed the people at the college. I wrote a program that does the grades, and it only takes three hours, and it only takes one pass. Instead of putting the cards through many times, one time.

And they said, we don't believe it works. So, why not? We don't believe it. Just trust us, it can't work. So, we did a test. We did their program, 12 hours. Then we did my program, three hours and one pass, and it came out exactly the same. It worked. So, why didn't they think it worked? Well, a year later, just as I was about to leave the college when I graduated, I found out why. They asked me to clean out an old file cabinet, and I came across a consulting report. The college had hired IBM, the biggest computer manufacturer of the day. It was 95% of the computers worldwide were IBM computers. So, it was the big company. They had paid them $3,000 to solve this problem, make the program run in one pass in a few hours. Experts at IBM wrote a report saying it's impossible. It can't be done. The computer is too small. By the way, the college paid me for my program. Remember, they paid $3,000, which in those days was a huge amount of money. It was two-thirds of what my father made in a year. They paid me $25 for my program.

So, I said to myself, either I'm really, really good at writing computer programs, I can do something that IBM thought was impossible, or the world of computers is filled with scammers. They can write a report and charge $3,000 just to say it can't be done. So, I should go into computers and either I'll be really, really good, or I'll make a lot of money just writing reports saying it can't be done. Can't be done.

OHI: So, was that your first program?

DC: Oh, no, not my first. That was the first big program where I had a real insight.

OHI: So, that turns out that you are really, really good at computer science.

DC: Yes, it turned out I was really, really good.

OHI: So, what's your first program?

DC: Oh, I don't remember. Oh, I do remember. One of the things that I was learning about was binary representation. And I kept asking myself, I have an integer, 15, any integer, and I want to know what is it in binary. I kept asking myself, how can I figure out this in binary? Now, the good news was I was a math major, so this was just math. And it took me a little while, but I figured out how to print out the binary value of 15, ones and zeros. So, I wrote a program to do that. That was one of my very first programs.

OHI: So, after you graduated from the college in 1971, you decided to continue your study at the computer science. So, why did you choose Penn State University?

DC: I chose Penn State for the stupidest reason of all. Remember, this was back in the day when computer science was brand new. And it was also in a day long before the internet where you didn't get a lot of information about the real world out there. No one at Houghton College knew anything about computer science programs. Where should I go to grad school? Nobody knew. But I found an ad that said Penn State had just gotten the biggest, most advanced computer that IBM had.

And, all right, I was a kid. I thought, if they've got the most advanced computer, they must have the most advanced computer science program. So, I went to Penn State.

OHI: So, have you disappointed about the fact that Penn State wasn't the greatest and advanced university in computer science?

DC: It's really hard to describe how sheltered and naive I was as a kid. I had no idea. No one had told me you should, you know, you should keep looking around for a while. You shouldn't just pick a place. And when I got there, I had no idea. Maybe all computer science departments were like that. So, I was very fortunate, though, there were some good people that showed up just when I did. Some young faculty were fairly pretty good. And they were studying algorithms at the time. So, I went through the program. I excelled. I learned everything. And I learned a lot more than most of the other students. In addition to my classes, I kept studying everything I could about computers and computer programs and languages. And so, I learned a bunch of things. Maybe I wouldn't have learned that if I had gone to some place that had a more rigorous program. Maybe I would have just studied what the professors had me study. You never know.

OHI: So, at Penn State University, was there any professor that made you realize that he was the one that I should learn from and learn after?

DC: Well, here's the thing. I have never had a hero. I've never had a hero who was everything I wanted to be. But I've had lots of people who had some particular talent or skill, and I wanted to emulate that talent. Let me give you a non-computer science example. When I was a kid in high school, I read F. Scott Fitzgerald. And I thought it was some of the most beautiful prose I had ever read. Unbelievably well-written English prose. But, if you look at F. Scott Fitzgerald's private life, it was a disaster. In many ways, he was a jerk. So, I wish I could write as well as F. Scott Fitzgerald, but I certainly don't want to be F. Scott Fitzgerald. And I felt that way about a lot of my professors in grad school. I wish I could prove a theorem as well as this person does. I wish I could grasp new topics as well as that person does. But there was no single person that I said, that's the person. Everything about that person is perfect, and I want to emulate that person.

OHI: So, you were just trying to learn from the best and try to be their best part.

DC: Exactly. Take the best part of everybody you meet. See if you can somehow live up to that.

OHI: That's a really very hard job, actually.

DC: But isn't that what you're doing all your life? When you're a little baby, aren't you looking at your parents and saying, “Wow, they can do all sorts of things I can't. I should learn how to do that.”

OHI: That's right. But I mean, for most people, they just try all their best to be best at one aspect. Maybe one major or one career or one aspect of life. But you are trying to be best in all aspects of life.

DC: What can I say? When I was in college, I took a course in English literature. Now, for a physics major, I wasn't required to take it. In fact, the only people in the room were English majors and me. Most of the English majors wanted to become writers. They were studying literature because they wanted to become writers. A couple of them said that they would settle to become journalists. But, you know, their goal was really to write great literature. So, I was sitting there. It was one of the best courses I had because it challenged me. Of course, you're thinking, well, how could you compete against all those English majors? And I couldn't. I didn't get A's. The most thing that I was proud of in the whole course, I got a B plus on an essay. And I thought, here I am competing against real people that want to become real writers, and I got a B plus. I learned a lot about writing. Now, if you look at my career, I have written a lot of textbooks. I get great accolades for having written the textbooks well. Lots of people tell me, this is a great textbook. This is well written. One person keeps saying, this is the best written textbook I've ever read. So, do you have to excel at everything? No, but if you can learn some good things about every aspect and every person you meet, you may not be the world's greatest, but you can be much, much better than you are.

OHI: Yes, you're right. Learn from the best and at least you'll be good. And so after you graduated from Penn State University in 1976, so immediately you go to Purdue University as a staff to work there.

DC: As an assistant professor, yes, I started right out of grad school. I had to choose between Bell Labs and Purdue. Bell Labs made me an offer. It was in those days we had these giant industrial research labs. Bell Labs was considered the premier industrial research lab. They had seven Nobel laureates. They had invented the transistor. They had invented the LED. They had invented all sorts of wonderful things. And they made me an offer and Purdue made me an offer. Purdue was ranked number seven in those days. And I talked it over with my wife and we decided we would go to a university.

OHI: What kind of reasons? Because Bell Labs was really the top one of the industry.

DC: Well, for one thing, I like to teach. And I have to tell you, I'm sort of saying this egotistically, but I'm very good at it. I get very good ratings. And for many years I had the top ratings in the department. I don't know what they are now, but… So, I like it. What can I say? We thought about it. We thought about a lot of things. And I decided the university would give me a lot of freedom. The other thing that if you're good enough in a university, you can branch out. You can start new things. You can look at new areas. I was trained, by the way, as a theoretician. When I went to grad school, most of the professors had a PhD in math. Theory was still very much the center of computer science. But I wanted to learn about compilers, operating systems, databases. I wanted to learn other things. A university gives you freedom to do that. And I took advantage of the freedom. I got to Purdue and I started studying other things other than what I was trained in.

OHI: And so how and when did you become interested in networking and protocol?

DC: One of the things I wanted to know about when I got out of grad school is I wanted to know about computers connected to each other. How do you connect them? What can you send? I was very fortunate to get a donation of some small computers and my second year at Purdue even though I was just an assistant professor, I hooked up these computers. I built software to send messages around. I did some, remember I had done some electronics when I was a kid, so I soldered together the wires and hooked up the connections between them and it was an incredibly slow, these computers were really old, they were really, really slow. They were being given away because they weren't any good. But I hooked up some computers and made a little tiny network. Now it was kind of silly. I mean it was just more or less a homebrew network, I never had any training in networking. But some people wanted to have a network for computer science departments and they were going to put in a proposal. The head of the department at Purdue was one of those people and he said, well, you've done networking. You should be on this project. And I said, I don't really know, I'm not an expert in networking. He said, what have you done? And I said, well, I built this tiny little protocol to communicate and send messages among these computers. So, he put me on the grant and said, you're going to be writing the protocol section. Write this proposal for how to do protocols.

And NSF didn't have enough money. So, we started writing a proposal. NSF didn't have enough money for our proposal. They went to DARPA and DARPA said, if you somehow use TCP/IP protocols, the internet protocols, we will help support this. Now the internet protocols were being developed at the time. They were just brand new. They were not working. They weren't implemented. So, I had to learn about the Internet protocols and how we would use them.

So, I studied the internet protocols. I came up with a way that I thought we could use them. It's a method called tunneling. And I went to a meeting at DARPA. I had to go there and present my ideas and see if Vint Cerf and the internet protocol would approve them. I got there. There was another well-known guy in networking there. And he said, what are you doing? And I said, I'm going to propose this tunneling. He said, no, everybody's doing protocol translation. Tunneling is a bad idea. You're going to lose. I'm going to win. So, he went in first. He came out. He said, it went very well. There's no point in you even telling your story. I went in. They asked me about protocol translation. And I said, I've looked at it. And I don't think there's any way to do the translation between the two systems we're using. It was TCP/IP and X25. Maybe there is, but I just don't see it. And I do see that I could make tunneling work. They chose me. So, we finally got the NSF proposal through. I did the tunneling project. It worked. The people that were doing network translation never succeeded.

OHI: So, have you ever got any response from that already famous networking expert about him losing and you winning?

DC: I saw him a few years later and he said, I'm still working. And I am going to show you. Someday I'm going to show you. Years went by and I never heard from him again.

OHI: Is it possible for you to let me know who was that?

DC: I don't think that would be good. You may be able to figure it out if you look at who was working in those days on protocol translation. But I don't think it would be kind. That was a long time ago.

OHI: It's still a part of history. We'd like to record it.

DC: Well let me say that a lot of people were working on protocol translation. For whatever reason, it was the acceptable solution. Everybody believed that that was the way to go. When I was an outsider in the community, I didn't have that sort of religion that said you must use protocol translation. I looked at the problem and just analyzed it and asked, how can I solve this? And I think that was the difference. Sometimes the community gets involved and invested in a certain way of doing things. And then it's really hard to break out of that.

OHI: Yes, sometimes it's hard for them to think out of the already existed box. So, everyone thinks the same way and there's no way for them to be out of it and to have another idea of how to make it work. So, speaking of TCP/IP, I've been told by someone that the only reason that it was chosen over OSI is because it was working, it works. And actually it was just a manufacturing job. It cannot be described as a technique or a science.

DC: All right, so here's what I'll say. The people working on the Internet Project are the brightest people I've ever worked with. Now, remember, I'm a PhD and my community is researchers with PhDs. These people stood out. They were incredibly bright and incredibly hardworking. One of the things that they did was they boiled down a really hard problem and found an elegant, small solution. The IP header has 20 octets, 20 bytes of data in it. And way back in the 1970s, people were saying, oh, we're going to need hundreds of bytes of data. They did it in 20 bytes. Same thing with TCP. The header is incredibly small. Now one of the mistakes that people make about engineering, and it's a universal mistake, it's not just computer science or computer networking, they look at a solution and they say, oh, that's so simple, anyone could have done it. And in fact, it's exactly the opposite. In engineering, it's easy to build a big, complicated, over-featured, you can keep adding stuff easily. The hard part is getting rid of things. The really hard part, the intellectual part, is boiling it down to the bare necessities. Engineering minimalism. And if you look at people who do architecture, there's a famous quote about architecture. Pascal once said, please forgive the length of this letter. I don't have time to write a shorter one. It's true. Making things small and elegant is hard. And so people have been saying that about TCP/IP for a long time. The OSI folks came from a world in which you just sit down with a committee and do engineering by committee. And that means that you do a compromise to put in everything everybody wants. Hey, I'd like to do it this way. Oh, I'd like to do it that way. Okay, we'll have an option. The first bit will say whether it's A or B. And then the guys doing B say, but within B, there's several different ways we could do it. There's three ways. Okay, we'll just add an option and we'll have B1, B2, and B3. And pretty soon you get more and more and more cruft added to this design. If you look at the way the telephone system has been engineered, it is incredible how much documentation and features and standards there are to do simple telephony. That's because it was all done by design and it was designed by committee. And in the case of telephones, it's international. Every country gets to have a representative to the ITU and do the design that way. That was the world in which OSI came along. The OSI folks thought what we need to do is get a committee. They'll have a bunch of good ideas. We'll put all their good ideas into our protocols. And then they made some silly assumptions. One of the worst ones was they were going to do something that was very equivalent to TCP. They called it TP-4. And they looked at TCP and TCP has a three-way handshake to start a connection. They said, well, we don't need three ways. We'll just do a two-way handshake. Send a message, get a reply. We're done. Everybody's happy. It works. Now what they didn't understand, somehow they missed the fact that a three-way handshake wasn't put into TCP because it was just extra cruft. It was necessary and sufficient. It was the minimum that you need to have reliable communication in a world in which packets can be lost, duplicated, delivered out of order, delivered with huge delays. You take all those things, a three-way handshake was necessary and sufficient. But you know, the OSI committee could just sit and vote. Let's vote. How many people think three ways is too much and we're going to be better than TCP by only having two? Everybody raises their hand. Okay, we'll vote it in. Good.

So, the reason that OSI was a bad design was it was a committee design. It was over-featured. It was incredibly inefficient. And everyone who implemented it realized how terrible the design was. And I've always told people, if anyone, OSI or anybody, has a great idea that can replace TCP/IP, all you have to do is write it up, put it on the table, and the world will give you billions of dollars. Do you know how much money is invested in TCP/IP? If you could do better, you could get billions of dollars in an instant. But the OSI folks would rather engage in, I don't know, slander and politics. There was a student at another university who contacted me in the last five years. He was writing a dissertation saying, isn't it true that the only reason TCP/IP won was politics? That somehow you guys got the world to accept your political view? And I told him the same thing I just told you, that if there was a better way, people would pay a lot of money for it. All you have to do is show it, prove it, build it, and show that it competes with TCP/IP. But everybody who keeps trying keeps losing. They actually implement their ideas. They don't quite work as well.

TCP/IP was a work of genius, the genius of a handful of people. And it's lasted all these years.

OHI: Were there some kind of challenges and obstacles you faced when you were working with TCP/IP protocols?

DC: Well, let's see. The first thing you have to understand is that in those days, the network hardware was horribly unreliable. The network links were slow and were horribly congested. Packets were lost all the time. Packets were duplicated all the time. I have saved a couple of things. One of them is I saved a note from Jon Postel, the late Jon Postel. I was talking to him one day. And we were trying to figure out what was going wrong. And it turned out that when I did a pin to ISI, where Jon was, from Purdue, almost every packet was duplicated, but not just once, five, seven, 12 times. Imagine building a communication system where the underlying hardware is screwing up badly.

So, the challenge was to make something work with horrible hardware. Now that sounds like a bad idea. In fact, why don't we just build good hardware first and then build protocols that work on it? And it turned out that having horrible hardware exposed all of the things that we needed to contend with because that's what could go wrong at some point. And once we were able to make protocols work with horrible hardware, then when hardware got better, they worked much better. I know it's sort of counterintuitive, but if you start with limited systems, you do much better than if you start with the world's greatest high-speed big systems.

If you're doing engineering and somebody gives you the highest speed processor, the largest amount of memory, the highest speed disks, the largest amount of disk space you can have, the most impressive high-speed communication system, you will build that ugly software. It's true. If the hardware gives you everything you need, the tendency is to just not worry too much about software. That's how we get the kind of systems we have now.

OHI: So, speaking of operating system, you have one called Xinu system.

DC: Yes, I did an operating system called Xinu. I told you when I got to be a professor, I wanted to try new things. And I always wanted to understand operating systems, and no one seemed to be able to explain operating systems to me, including people who were teaching operating systems.

So, I thought the best way to do it was to build one. So, I built a small operating system, and I'm very proud to say that it has been used around the world ever since.

In 2021, a friend of a former schoolmate of my son's contacted him and said, she's an audiologist. She was at a meeting of audiologists. They're people that do hearing, nothing to do with computers. But she said there was a new hearing aid. And this new hearing aid was running an operating system called Xinu. It was done by a Professor Comer at Purdue. And she asked, she knew my son was, you know, she knew I was a professor. She knew my son's name was Comer, my name's Comer. She asked, could this be your father?

So, in 2021, Xinu was used in a hearing aid. And by the way, if you haven't seen modern hearing aids, you need to look at them. They contain two protocol stacks, plus all of the software to do digital processing. The two protocol stacks, here modern hearing aids have built in Wi-Fi. So, they've got complete Wi-Fi protocols. And if you go to a concert and you're hearing impaired, you can get the concert audio in your hearing aids over Wi-Fi. They also have a short-term Bluetooth connection so that when you go to a doctor, you go and take your hearing aids in, you wear them to your doctor, they can analyze your hearing by controlling the hearing aids from their computer system while you're wearing them. So, fairly sophisticated software running in a hearing aid.

OHI: So, that small operating system you developed almost 40 years ago, still working in very broaden areas. So, why did you decide to call it Xinu?

DC: It actually stands for, it's a recursive acronym. It stands for Xinu is not Unix. Students would ask me, your operating system does it this way, but Unix does it that way. Why didn't you do it Unix way? And I said, well, first, it was a very small system.

I was working on an embedded system, so I wanted it to be small. But secondly, I just wanted to see if you could do it a different way, and how it worked, how it compared.

OHI: So, just to find out another way to do that, you invented the Xinu operating system. And so what were some of the key goals and objectives you had in mind when designing that system? Besides you want to have another way to go?

DC: Well, I wanted to understand operating systems. Remember, I was a researcher, and it was really a research project. I wanted to understand the relationship between interrupts and processes. Interrupts is a hardware idea. Processes is when you run an application on a computer, you run a process. It's an operating system abstraction. And I kept asking people, what's the relationship between interrupts and processes? And no one could give me a good answer. And I wanted to explore that. So, that's one of the things that I learned as I was studying Xinu, trying to figure it out. Another thing that I wanted to understand was, how does structure operating system code? And I noticed that there were several different things an operating system does. It has a file system. It has this processor idea, process management idea. So, it has to do process management. It has to do processor management, that is to say hardware. It has to handle interrupts. How do you structure all this to make it work together?

And I asked myself, could I build multiple levels of an operating system? Could I start with a process manager level and add on to that process communication and add on to that and just keep going up process coordination, message passing? Could I build everything in a strict hierarchy? So, it was a research question. And the answer was yes, I could build things in a strict hierarchy. And what are the principles that you need to follow to do that? So, I developed principles for how to build this hierarchy, how to handle interrupts and interrupt masking in a hierarchy, how to handle process management, and of course, how to put in internet protocols. Internet protocols all run inside an operating system.

OHI: So, what's the biggest challenges you faced during your development process?

DC: The biggest challenges. One of the things that was not a technical challenge, but it was a challenge that I had to face. Many people didn't think it was good to build operating systems, to do internet protocols, to work on the internet project. Many of my colleagues, including the Department Head that I had during some of the internet project, they were very down on it. They didn't think I should have computers. They didn't think I should build a network. They didn't think I should build an operating system. And they tried to get my projects banned. They tried to say, we want to make sure he doesn't do this. Let's make sure he doesn't accept any gifts of computers. Took me a long time to figure out that I was threatening people. I thought, remember I was fairly naive when I became a professor. I thought everyone would be just like me. They would want to explore everything. But a lot of professors, I guess a lot of people are sort of frightened when they take a job. They don't know if they're good enough. So, instead of wanting to branch out and explore things and try new things, they want to make sure that they act on their jobs so they're doing the same thing that they did yesterday. They can keep repeating the same stuff they're doing. There's never any challenges, there are never any new courses to teach because, you know, they can teach the same old stuff.

And by me coming along and saying, hey, we need new courses, we need new projects, we need new areas of computer science. I was threatening them. And being a nerd, I was not especially good with people. I didn't understand that I was threatening them. So, instead of sort of consoling them and reassuring them, I took the opposite approach saying, “Well, why don't you try something new? You'll like it.”

OHI: So, it's kind of like office politics, it's hard to deal with all those kind of persons in the office.

DC: Yep, it's hard to deal with them. And I once read a paper that said there's a fundamental error in psychology. The fundamental error is when you meet someone, as long as they seem roughly like you, they have roughly the same education, roughly the same economic status, roughly the same, you know, background, you assume that they are just like you inside. And it took me many years to realize that I was not like most people. I was not like most professors. And therefore, I couldn't judge what they would like, what they wouldn't like, what threatened them, what made them feel good.

I'll give you another example. About 10 years ago, there was a committee assignment here at Purdue or somebody needed to write up this big document. And I looked at it and I thought, oh, what a waste. That document has no real science in it. That's just how we're going to set up this course and all the little procedures and how many, you know, credit hours and this and that and the other thing, all these little details. What a silly waste. And the Department Head said, I'm going to have to assign this to someone. And I thought, I hope it isn't me. And I assumed everyone in the room was saying the same thing. Inside they were saying, I hope it isn't me. I hope it isn't me. And the Department Head picked someone and said, I would like this person to do it. And I thought, oh, that poor person got it. It turned out the Department Head had an advantage that I didn't have. He could read people. He read this faculty member, gave him this job. And then the faculty member for the next six months got up and said, I've been working on this big document that I'm so proud that I've gotten the first 10 pages done and now I've got to do 35 pages more. And we're doing really great stuff. A faculty member on my faculty enjoyed it. It's really hard to take yourself and ask, what do other people enjoy? What do other people aspire to? They're not like you. Not everyone is the same as you. And that's a hard thing to do. I find it hard. But I will say there are people who are good at guessing that stuff. They can read others. And they can say, that person is feeling bad. And here's one way to make them feel better.

OHI: Yes, that's kind of a talent too.

DC: It is.

OHI: To read people. So, they are kind of like person`s person. They know person and they know how to deal with them. And they know how to make good use of them.

DC: Right. If you're a good manager, you have to have that skill. You have to be able to read people. I took a leave. I was at Cisco for a while. I was their VP of Research, first VP of Research. And I saw a manager, and one day he called me in and we had a discussion about what future projects ought to be done and how research can help. And I walked out of the office thinking, he's a technical guy. He knows exactly all the technologies and he really is good. Then another guy went into his office and came out and said, boy, this guy agrees with me that we don't need any new technologies. What are you talking about? That's all we talked about was the new technologies and how he wants to move forward. He said, no, no, no. He told me no new technologies. And let me tell you, that was a great decision because I think new technologies are a scam. We shouldn't be doing that. That manager was so good that he could look at people and convince them that he agreed with them and he could agree with completely opposite views.

OHI: So, what's exactly their own view?

DC: Well that's the other thing about a manager. A manager has to be able to look at everyone around him and figure out who the good guys are, who the people with real good ideas are. And that's very hard, especially in industry, but even in academia, it's hard because so many people will tell you how great their ideas are.

And I have found that the best scientists are never braggarts. The best scientists will tell you, here's an idea and it works really well in the following circumstances. We haven't tested it in this circumstance, so we can't say anything about that. But then you meet some terrible person who's not very good and they will walk in and tell you, I've got a great idea. Why? It's so fantastic. It works in all cases. It's faster, better, cheaper, less expensive for the long run. It will save everyone money. And if you're a manager and you don't know technology, it's really hard. You have to be able to separate the really good people who deliver really good stuff and are quiet about it from the braggarts who make exaggerated claims about what they can do and what their stuff does.

OHI: It's really a very hard job. So, that's why only some of them can be a good manager. Most of managers are not suited for their jobs. So, back to Xinu. It was designed to be a highly efficient and lightweight operating system. So, how did you achieve this? And what were some of the trade-offs you had to make in order to optimize the performance?

DC: Let's see. How do you achieve things? It was once a Nobel Prize winning scientist and he was asked, how do you come up with a solution? And he said, there are three steps. You write down the problem, you think very hard, and you write down the answer. And for Xinu, so I had to think very, very hard. How can I build this in a way that doesn't duplicate code? So, it had to be small and it had to be efficient. And one of the things that I observed in an operating system was there are lots of duplicate code areas. For example, processes get put on lists. They get put on a list of sleeping processes. They get put on a list of ready processes. They get put on a list, on and on and on. They get put on a list of blocked processes. And most operating systems had multiple mechanisms. It was each of these things had a queue or a set or some data structure to hold lists of processes. And so one of the things I asked is, can we have one data structure that holds all lists of processes? I worked on it. I came up with it.

How can I make this thing smaller? How can I make my data structure very small? One of the things was relative pointers, an idea that doesn't get used much because these days we have giant memories. And so everybody has big pointers and they just define a pointer without even thinking about it. But I was asking for a very, very small machine, how can I make pointers really small? And the answer is, don't use pointers the way we normally use them in a computer program in the address of memory. Use an index into an array. An index is a relative number. Indexes start at zero: 0, 1, 2, 3, 4. So, instead of having a pointer to an arbitrary memory location, just have an index. And now you can make the index really small. Only as many bits as you need to accommodate the total number of processes you need.

So, there were lots of little decisions like that throughout Xinu. How can I work to make this small and as elegant, avoid all duplication? So, no duplicate code, no duplicate data structures.

OHI: So, how has Xinu influenced the development of the other operating systems?

DC: How has Xinu influenced the development of other operating systems? Most operating systems these days were all influenced by one operating system, Unix. The people at Bell Labs had built Unix. When I got out of grad school, they wanted to hire me to work on Unix. It was by far the biggest step forward in operating systems that had been made. And if you look at Mac OS, what is Mac OS? Guess what? Underneath it's Unix. If you look at Linux, what is it? It's just a repackaged Unix. So, everything was influenced by Unix.

Will I have any influence at that level? Probably not. But there are lots of little embedded systems that have used Xinu or variants of Xinu. And there are people who have taken Xinu and modified it for their embedded system. So, in the embedded world, it has been influential. In fact, parts of Xinu code have been to Mars. There's a company named VxWorks. VxWorks sells the most widely used real-time embedded system software. It's been used by NASA. And VxWorks was derived from Xinu.

OHI: So, what do you think are some of its most enduring contributions to the field of computer science?

DC: Enduring contributions of Xinu. Well, it's done two things.

One, I wrote a book about Xinu. I hear from lots of people who say, I was trying to understand operating systems. I found your book. I read it. And now I understand how an operating system really works. Some of these people, by the way, have worked on real operating systems like Windows. So, these aren't just students who are, you know, in a course somewhere. They are engineers. And I think having Xinu so they can see it and understand it is important because you can understand Xinu in a semester. 20,000 lines of code. It's nothing. Compare that to trying to understand Linux. 8 million lines of code. You'll never get through it. You'll never understand all the pieces that are there. So, I think it has helped a lot of engineers understand the principles and the basics. And then they can go on and work on real systems.

OHI: And if you look back on the development of Xinu, what are some of the things that you are most proud of? And what are some of the lessons you've learned from the project?

DC: Well, let's see. I already told you that the table that keeps lists of processes is a very interesting idea. In addition to using indexes instead of pointers, it uses an implicit data structure. I'm very proud of that. There are very few algorithms that have an implicit data structure.

What else? There is an idea in Xinu that I think every CS student ought to see. It's the notion of a syntactic namespace. In most operating systems, file names and directory names are inherently hardwired into the file system code. In Xinu, there is a file system. Separate from the file system, there is a naming system. And you can define names. You can define, for example, remote and local files. And you can make them correspond to local and remote file systems. And you can do that without hardwiring it into the file system. So, that's an interesting idea. It also has this idea that there is a relationship between syntax and hierarchies. If you think about a file name /x/y/z, you can treat it syntactically, in which case some of the prefixes correspond to directories. So, there's that relationship between prefixes and directories. The Xinu syntactic namespace does that. It shows the relationship.

There's also one more thing in Xinu that I'm very proud of. There's a module initialization system. It's called memory marking. When I was in grad school, I was studying algorithms. And one of the questions was, can you do constant time array initialization? On the surface of it, it looks like you can't. You know, you have a big array you'd like to initialize. You have to initialize the whole thing. And then you can insert values and pick them up. But there's a way to do it. There's a trick that you can use. There are a couple of tricks you can use. And in grad school, I figured out how to do it. Not only did I figure it out, but I figured out a way that was different than the way the textbook guys had figured out to do it. And I noticed that it was very practical. And when I was building Xinu I realized I could do a memory marking system and do module initialization and use this algorithm that I developed in grad school. So, I did. So, the memory marking system in Xinu is very, very nice. You just pick a location in memory and say, I'm going to use this as a memory mark. And you can ask, has it been initialized? And the answer is no. And now you stick a value in it. Has it been initialized? And the answer is yes. And there's a function that asks, has it been initialized? Has it not? And the memory marking system works in constant time. In fact, there's only a few instructions needed for each answer.

OHI: And also, do you see any opportunities for Xinu's further development or the use of it in today's industry?

DC: Who is the user base? Is that what you're asking?

OHI: Yes. And also, besides today's usage of Xinu, what else can use Xinu for its development?

DC: Well, it's really the embedded systems world. You know, all the IoT devices, internet of things, all of those devices have a little operating system in them. And if you want to have minimal amounts of memory, minimal amounts of processing, you want to have a very small IoT device. And by the way, the reason you want to do that is minimal amounts of memory and minimal amounts of processing power mean longer battery life. So, if you're trying to design a system that has very long battery life, you need a very small elegant operating system that doesn't take a lot of space, doesn't use a lot of memory, doesn't use a lot of processing. That by the way, is why they work in hearing aids. One of the goals is have as long battery life as possible with very tiny batteries. So, that's where Xinu is likely to be used more than anything else.

OHI: So, it still has a very large area to be used.

DC: Yeah, well, it still has a very large area to be used in. There are more embedded systems in the world by far than normal computing systems.