John Backus
Video Interview Transcript

Interviewed by:
Grady Booch
September 5, 2006, Ashland, Oregon

GB: Grady GB (Interviewer)
JB: (John Backus, interviewee)

GB: I'm here today with John Backus and I should probably put the day's date [September 5, 2006] on here for the calendar, for the purpose of filming. But there's probably more computational power in this Sony camera than there was in the first computer upon which you worked.

JB: Oh, I'm sure.

GB: You are a person whose career has spanned the ages of contemporary computing, from the earliest days of some of the first machines that really made an impact upon the commercial world, to where we are today. Of those years, what do you think surprised you the most, of the changes you've seen?

JB: Well, I think just the speed of change. I mean it's just appalling.

GB: Appalling.

JB: Yes, because I mean all these refrigerator-sized machines that I first worked with soon became smaller and smaller and smaller until--

GB: In fact, as I read some of the interviews, you're a fan of the Palm Pilot and I think you had a TiVo at one time.

JB: I still do.

GB: Still have a TiVo, which probably did have more computational power than the first machines upon which you worked.

JB: Yes.

GB: What I heard you say is just the first derivative of change, that things are changing so rapidly. How wired of a guy are you? I mean you have your e-mail address. How connected are you these days?

JB: Not a lot. I'm like any guy that has a personal computer, fumbling around trying to find my way in this mess that's out there.

GB: What was your first personal computer?

JB: It was an IBM PC.

GB: I imagine you got a good discount on that.

JB: This big, you know, this high. You could hardly lift it.

GB: Wow. Amazing. In fact, let's go back to some of the machines over which you've worked over the years because the first one was the SSEC. If I get that right that's the Selective--

JB: Selective Sequence Electronic Calculator.

GB: Yes. What was the story behind the existence of that very computer? Why did it come to be?

JB: Oh, that was because IBM cooperated in designing the Mark I at Harvard, and Harvard gave them no credit. So Watson was really mad about that.

GB: That was Watson, Sr.?

JB: Yes. So he decided to build this strange monstrosity called the SSEC. And he did.

GB: Now was that the defense calculator?

JB: No, no.

GB: Was that a different name for it?

JB: The defense calculator was the 701, which followed.

GB: Oh, okay, which was the successor to that.

JB: Yes.

GB: Where was the SSEC actually built?

JB: It was built right there on 57th Street-- Well, the entrance was on 57th Street, yes, but it was near Madison.

GB: I don't think there's a lot of manufacturing in downtown Manhattan these days of computers.

JB: No.

GB: That's really what attracted--

JB: It was unique.

GB: It was unique, one of a kind.

JB: Yes.

GB: You were describing earlier what that beast looked like. It was many rooms full.

JB: Well, yes. It filled a room that was about, let's see, 50 feet wide by 100 feet deep. It had a window opening on the street so that people could look in, and big black columns holding up the middle of the room. It had a huge console with hundreds of toggle switches and stuff like that. Then behind

glass cases around the walls were tape units and relays, some of which would fall out as it was operating because they were heavily used. It was a fun machine. It would make an error about every three

minutes and you had to stop and figure out how to restart the thing.

GB: Right. That was certainly the age of machine building, because there was the Mark I around the time.

JB: Yes.

GB: That was Aiken, if I'm not mistaken.

JB: Yes.

GB: And Eckert and Mauchly. That was around that time or a little bit earlier in their work?

JB: That was about the same time.

GB: Did you have any contact with those guys at all?

JB: No.

GB: Just a quick side story. I think it was Mauchly's grandson [who] went to the Air Force Academy, and I taught him COBOL, of all things. Mauchly came up once to visit us. It was interesting seeing him. I recall [in] one interview that you were walking by the IBM offices and saw this display, and that's what attracted you inside then.

JB: Yes.

GB: What were you doing at that time? You were out of the Army?

JB: I had just graduated. I had just gotten my Masters Degree from Columbia. But I hadn't even begun to look for a job, but I just found this place and I walked in and it looked so interesting. I ask if they would give me a job.

GB: And they said?

JB: And they said, "Yes, come up and see the boss." I said, "No, no, no. I've got holes in my sleeves. I have to look respectable." But they got me up there anyway and I got an interview by [Robert R.] "Rex" Seeber who gave me a little puzzle that I solved, and he hired me.

GB: Wow, on the spot?

JB: On the spot.

GB: My goodness. So at Columbia, you were getting your Masters in Mathematics. Is that correct?

JB: Yes.

GB: Actually, to go back in time a little bit, before that, before Columbia, you were in the Army then.

JB: Let's see, yes. Yes.

GB: So that was really the time of the Korean War or World War II?

JB: That was World War II. That was 1945.

GB: Oh, my goodness. So let's start there then. Where were you stationed in World War II?

JB: Well my first station was in Camp Stewart Georgia.

GB: Oh, my - hot place.

JB: It's a very bleak, hot place. But I only stayed there for a few months, and then I was selected for army specialized training. Then I went to some little place in Alabama to get classified. Then I went to the University of Pittsburgh, which I enjoyed enormously because I was supposed to be learning all this stuff that I had already studied. So I spent a lot of time there. There's a place called the "Pittsburgh Playhouse," which was really a bar. A very friendly place. After that, yes, I was there. Then the Battle of the Bulge occurred and they were drafting everybody. So everybody in Pittsburgh was going to be sent overseas right away. But I had just taken a test, some kind of a test that qualified me for being sent to Haverford College. So I went there and had a very nice time at Haverford College.

GB: Where's that located?

JB: It's just outside of Philadelphia in Haverford, PA. I was there for a while, and, let's see. Then my next step. I was destined to go to medical school, but in the meantime, I was sent to Atlantic City to work in the hospital. So I stayed in Atlantic City working in this hospital 12 hours a day for a while, living in the Traymore Hotel on the Boardwalk. While I was there, I was working on the neurosurgery ward and they noticed that I had this huge bump on my head here, which was a bone tumor that had been growing slowly, fortunately outward, for a long time. They said, "Well, let's take that out." So I became a patient. They operated, took it out and I was a patient on this ward. So I had no duties whatsoever. I could just spend my time wandering around Atlantic City at night, which I did, and sleeping in the hospital. That went on for a while. Then finally I got sent to New York, and started going to medical school at - what was the name of it - Flower and Fifth Avenue Hospital. I discovered very quickly that I didn't like medical school, because all you had to do was memorize stuff. While I was in Atlantic City, they had done this operation, removed this bone tumor and put in a plate, which they had constructed by sort of cutting out triangles, because there's a lot of curvature in the

skull there. It was kind of a squishy plate, which made me feel very insecure. So when I was in medical school, I opted to go to a hospital on Staten Island where their business was putting in plates for veterans. I went there and they took out the old one. I walked around for a while with nothing, just skin.

GB: Oh, my. That's exposed.

JB: Yes, it really felt weird. Then they took a cast and they let me make the plate, actually, because you just had to put it in a hydraulic press and trim it up a bit, which I did. They installed it. During that time, I got this wonderful little apartment on East 71st Street in New York. It cost $18 a month.

GB: Oh my. That must have been an exciting time to be in New York City, in Manhattan.

JB: Yes, it was. It was a Hungarian neighborhood. I'm sorry, I mean Czech, because there was a Czech restaurant just down the street from where I lived. It was very nice. No, I lived there for quite a while until finally, I got married. But I had some friends from the army who lived nearby. One guy was a composer who lived with his wife on the floor below in the same ratty apartment building, and another friend who was a singer lived across the street. So we had a nice little--

GB: So in walking by the IBM building, you saw this machine and then got whisked upstairs.

JB: Yes.

GB: Remarkable. Did you have any interest in computing prior to that time?

JB: No. I mean… No.

GB: There wasn't a lot of it around.

JB: What was computing? What's that?

GB: Fascinating. If I may ask, let me pick up on something you said.

JB: I just like machinery.

GB: Yes, because you're kind of a gadget kind of guy. How did you meet your wife, if I may ask?

JB: Well, I was married twice. My first wife; how did I meet her? Oh, I met her at-- She was living with three other girls in an apartment in New York and this guy I had known at the University of Virginia took me to this place. That's how I met her. My second wife I was introduced to by my first wife at about the time we were about to split.

GB: Fascinating.

JB: Because Barbara is an English teacher at Berkeley, and Barbara had taken her course in poetry.

GB: Small world. So back to the SSEC. What did they hire you to do?

JB: Programming, what else?

GB: Down at the machine level. What did "programming" mean for a machine like that? To a contemporary programmer, that would be so foreign.

JB: Yes, oh, I mean it was weird because you had the whole machine to yourself for months.

GB: Wow. I imagine the room got a little warm too.

JB: No, no. It was air-conditioned. The first problem I was assigned to was this one, that one of the machine's co-inventors was working on. It was a problem to calculate the position of the moon, which is a very difficult thing to do. It's a Fourier series of about 1,000 terms or something. I worked on that.

GB: This was long before the U.S. had committed itself to any space program really, actually any lunar program at all.

JB: No, such ideas were just totally-- No, I mean you're making a big transfer in time there to think in those terms.

GB: Interesting. So your background as a mathematician served you well, then, to do that kind of thing.

JB: Well, to say my background as a mathematician-- I was never a mathematician, really.

GB: Really.

JB: I mean, no. I like some of the more abstract stuff, but I was never a scholar. I never liked to study or learn anything.

GB: You mentioned one of the designers of the SSEC. Who was that person?

JB: Oh, Rex Seeber.

GB: So then from your experience with the SSEC, you then went on to produce Speedcoding, the Speedcoder.

JB: Yes.

GB: What were sort of the things that influenced you to create that in the first place?

JB: Well, programming in machine code was a pretty lousy business to engage in, trying to figure out how to do stuff. I mean, all that was available was a sort of a very crude assembly program. So I figured, well, let's make it a little easier. I mean it was a rotten design, if I may say so, but it was better than coding in machine language.

GB: Sure. So this was really around the time where we saw the explosion of languages like that. It wasn't really the first language above machine language.

JB: No, no.

GB: Who were some of the contemporaries around that time?

JB: Oh, God. I don't think I'm going to be able to answer that question.

GB: No worries. Because Grace Murray Hopper and the COBOL work, that was roughly around the same period as the early Fortran work, wasn't it?

JB: Let me see. Yes, I had a lot of—[pause]

GB: I forget when Grace's work actually was and the COBOL work.

Gardner Hendrie (cameraman): Yes, I think that may have been--

GB: It's a little bit later.

Gardner Hendrie (cameraman): It's a little bit later, yes.

GB: I still have my nanosecond from Grace. She had this thing where whenever she'd lecture, she'd always give people in her audience a nanosecond, which was a piece of telephone wire cut to 11 ¼ inches, which was the distance that light would travel in a nanosecond. She offered that as a visual metaphor to say this gives you an idea of why machines are shrinking. So if I think about the SSEC covering 100 feet, delays of the speed of light intruded upon things here as well. From the experience of the Speedcoding work, that's kind of what led you to write the memo to your boss saying, "Hey, I have this idea for a high-order language."

JB: Yes, because we were moving to the 704, which had built in floating point, built in index registers, which was all that Speedcoding was supposed to supply. So what the hell?

GB: And the 704 being the first machine with core memory as well too.

JB: Yes.

GB: Tell me a little bit about the 704 and its sort of size and shape and such.

JB: Well, it was a box about this big, about that thick and about that high. It was a big--

GB: It was a big machine.

JB: Yes.

GB: As we were talking earlier, the management of IBM was of mixed feelings with regards to the utility of the 704 because you had Tom Watson, Sr….

JB: Yes, he was very skeptical of it. But Jr. was not. He was sort of for going in this direction.

GB: Wasn't this around the time that Tom Watson, Sr. said something to the effect that the worldwide market for computers is like one or two computers, or something like that?

JB: Yes.

GB: He was saying that in reaction to the 704?

JB: Yes.

GB: Interesting.

JB: Well, first before the 704 was the 701, which had no index registers. It was a very primitive machine.

GB: Were you involved with the machine designers at all, or was this sort of just handed to you?

JB: No, I was part of the-- In fact, I sort of credit myself with getting index registers and stuff and floating point built into the 704, because the designers were just totally preoccupied in getting a drum unit designed. You know, one of these crazy magnetic drums.

GB: Which would have had how much storage capacity?

JB: Oh, about maybe 1,000 words, or maybe as many 10,000, but no more.

GB: Probably less memory than your watch has.

JB: Yes, right. Things were bigger in those days.

GB: So the market for the 701 and the 704, where was IBM trying to head with that?

JB: The 701, there were 18 701s, and the 704, there were quite a few more. I couldn't tell you how many.

GB: So you actually influenced the design of those machines then, the introduction of--

JB: Yes, I was on the design team for the 704.

GB: Was the notion of bringing index registers and floating point in viewed as radical by the design team?

JB: Actually, I kept sort of suggesting that they do this, and they kept talking about this damn drum. In the design meetings, I kept bringing this up and bringing it up. They kept talking about the drum. Finally, I decided, well, there's only one way to get their attention. So I spent an hour just deriving some cockamamie scheme for designing it. Gene Amdahl… It would have taken about a ton of hardware to implement what I had described and Amdahl said, "Oh, you don't need to do that." It's just easy to do it [with] just this. It doesn't take hardly any more hardware."

GB: Wow.

JB: He then designed it.

GB: Remarkable. So the interaction between you two led to some important innovations in that machine then. Did you do much further work with Gene?

JB: Not a lot, no. I mean, I was on some… IBM was trying to design something for the Defense Department, some big, huge – the NORC -- and we kind of interacted.

GB: Right.

JB: There were a lot of committees in Poughkeepsie, and stuff.

GB: So what kind of programming tools, as we would call them today, even existed for the 704 in its earliest days before Fortran?

JB: Well, there was an assembly program.

GB: Right, and that's about it.

JB: That was about it, yes.

GB: Wow. So from that experience, it led you to write a memo to, I think it was your boss Cuthbert…

JB: Hurd.

GB: Hurd, that said, "Hey, I have this idea here."

JB: Yes, I mean, you've got to make it easier to program this thing. I kind of laid out the fact that half the cost of running this thing was programming it. I mean, in counting the machine costs and everything.

GB: Right. In fact, you use a phrase in one of your interviews that said, "The assumptions under which we created Fortran really aren't valid anymore." What were the assumptions that Fortran was created with? I think you just said it, to some degree. It was reduce the cost of programming.

JB: Right. Yes, because it was just so slow.

GB: The process of producing programs.

JB: Yes. Machines were very expensive too. The rental for a 704 was in the millions [of dollars] for a year's rental for one of those things.

GB: Right. I recall reading a figure; it was like $400 an hour of computing time or something like that.

JB: Yes, that's about right.

GB: Who would think of charging per hour these days? I don't know how you'd compute that. But then the salary of a programmer would be a lot less than $400 per hour I would guess.

JB: Right, yes.

GB: So as you began the work into what became Fortran, what were the other things swirling about in the community that led you to that particular structure? Who were some of your contemporaries that were doing similar things? I don't even know some of the names around that time.

JB: Well, Grace Hopper was talking about compilers and stuff like that.

GB: Right. She really talked up what became COBOL.

JB: Yes, but her ideas were just so cockamamie stuff. Her scheme for this - I forget the name of her proposed compiler - but it was part machine code, part this, part that.

GB: Just out there.

JB: Completely unworkable thing.

GB: Right. She was working for Remington at that time?

JB: Yes.

GB: And IBM and Remington… Well, this was the era where there were a lot more computer companies than there are today.

JB: Yes, right.

GB: It was the age of the big iron where they were competing with one another. Had Gene Amdahl gone off at that time, or he was still with IBM, wasn't he?

JB: I think he was still.

GB: Okay. Interesting, but that was certainly a ferment of activities and ideas that spawned a lot of other companies later on.

JB: Right.

GB: Now, the first formulation you had of what became Fortran, that essentially came from your ideas. But then very quickly you began to assemble a team around you of about, I think it totaled 13 or so in all.

JB: Yes, but basically, the core of that was more like eight.

GB: Right, because you had Irv Ziller, as I remember, was one of the first ones that joined you.

Tell me a little bit about Irv and how you latched onto him and vice versa.

JB: Well, I was originally in what was called a Pure Science Department which Rex Seeber ran.

GB: This is really before IBM had started the lab, is that correct, or just around the same time?

JB: No, Watson Labs was in existence.

GB: Okay.

JB: So yes, Hurd was running the Applied Science Department, and Irv was one of the first people in the Applied Science Department. When I started working on this project, I got moved to the Applied Science Department because Seeber really didn't want anything to do with this stuff. When I pitched this to Hurd, it was easy to persuade him to let me get Irv to work with me. It just sort of went like that - one by one.

GB: That would have been in 1954ish- 1955?

JB: Yes.

GB: So you two toiled for a while and then slowly the team grew over time.

JB: Yes, we were allowed to hire people.

GB: As I recall, the management touch around you was pretty light, that perhaps--

JB: Very light.

GB: I'll put it this way, you probably succeeded because you were left alone.

JB: Yes. We were off in a building on 56th Street on the fifth floor of a little small building.

GB: Tell me about some of those years, because I guess it's sort of like a temporary software that you kind of set a date saying, "Well finish by then." It kind of stretched out, didn't it?

JB: Yes, it kept being extended by six months every time somebody asked. But we had a great deal of fun. It was a very nice group of people. My main job was to break up chess games at lunchtime because they would go on and on.

GB: In fact, I read that you guys used to play a lot of blind chess. Maybe I mistook it in one of the interviews, but tell me about that.

JB: Yes. Well, I don't know who… Harlan Herrick was the chess freak and he got people involved in that. There was some of this blind chess going on. But it was mainly just regular chess.

GB: Right. You've repeatedly used the words that that was a really fun time.

JB: Yes, it was.

GB: What made it so fun?

JB: Well, just because we all got along very well and we had these challenging problems to deal with that we kept talking about and discussing and we-- I don't know. It was a challenge.

GB: Sure

JB: The excitement of doing something that everybody said we couldn't do.

GB: Right. You were in your - what - late 20s? I imagine all of you were about the same age or thereabouts.

JB: Yes.

GB: I recall in one of the interviews, you guys sort of made up things as you went along, as you discovered the problems and would tackle them.

JB: Oh, absolutely. The problem kept sort of sub-dividing like an amoeba. The whole thing just got divided up into these phases.

GB: Sure, the phases of compilation, in effect?

JB: Yes. Each phase was worked on by one or two or three people, and they kind of conversed with each other to get the interfaces right.

GB: And your role was sort of overall architecture for it, it sounds like.

JB: Yes, my role was just to sit around and watch.

GB: [Laughs] So what were those major phases? How did it break itself out?

JB: Well, the first one was doing the arithmetic stuff.

GB: Sure.

JB: That took in all the data from the [source] code that was written, and it produced the arithmetic code and stored a whole lot of data for the next phase. The next phase was dealing with indexing and that they really couldn't deal with because of only having three index registers.

GB: Oh, my.

JB: So we decided that -- or I decided that -- they should do it for a machine with an unlimited number of index registers. And they did that. The third phase was just, sort of, just put together all this stuff that had accumulated. The fourth phase then did this sort of - what's the name of that - Monte Carlo calculation to determine how to assign index registers.

GB: My goodness. You were doing all of this in machine language or assembly language?

JB: Yes.

GB: Wow, I can't imagine doing that. It's hard enough in a high-order programming language.

JB: Yes. No, I mean these guys were really good.

GB: So when this first got unleashed to the world -- because IBM basically shipped this with every 704 then didn't they?

JB: Yes.

GB: Of course, it was bug-free in the first release.

JB: Of course. No. Apparently, we shipped one box of binary cards, which is the only one we were able to produce because it just ruined the card punch to produce these things. So we produced one box of binary cards and shipped it off to Westinghouse, somehow or another. And it arrived. Just this box of binary cards. They figured that this must be the deck for Fortran. They actually ran it, loaded it and then executed this program and finally had a compiler, which they then had a little test Fortran program which they compiled, and it ran.

GB: What did it do? Do you remember?

JB: Oh, it was some jerky little nothing.

GB: Right, but it was the first instance of an executable outside of your own group.

JB: Yes.

GB: Wow.

JB: Of course, that was the only time it worked for them for a long time thereafter.

GB: Did you spend a lot of time with them personally then, trying to get things working?

JB: We then sent this thing to a whole bunch of people and they all struggled with it and kept sending us all these error problem--

GB: Bug reports, yes.

JB: …error reports, and we kept correcting the errors. Slowly, I guess it took six months to sort of get it so it would run pretty reliably.

GB: Gee, that sounds how Microsoft delivers software these days. We probably don't want to keep that one on the tape. What were some of the early problems that people were trying to apply this to?

JB: Oh, they were mostly aerospace stuff. That was for the big customers of the day.

GB: Sure. Like Boeing and--

JB: Boeing, North American and - I don't know - the company that Roy Nutt worked for- GB: That name is familiar. Don't remember. Roy was a person on your team, wasn't he? JB: Yes.

GB: But what work did he do in the phases?

JB: He did the assembly program. It was a very special, fast assembly program that assembled all this stuff. But he did a lot of stuff. He came down from - my memory is so terrible.

GB: I might even have that in one of these things. [Looking at documents.]

JB: Yes.

GB: Steve Lore [Steve Lohr, a senior writer and technology reporter for the *New York Times*], who I met once, did a really interesting history of this work in one of his books – *The Turning Point*.

JB: Walter Ramshaw was his boss. It was an aerospace company, but it was in the east.

GB: East Coast. I'm not sure.

Gardner Hendrie (cameraman): Was it Grumman?

JB: No.

GB: Yes, I don't see the name here. You had one woman on your team. Lois?

JB: Yes, Lois Haibt.

GB: Yes. What role did she have in the midst of all of this?

JB: Lois - let's see, what section did she work on? I think she worked on section three. No, section four.

GB: We were talking about some of the other people involved, and the places where Fortran was making some inroads. It was primarily in the aerospace world, were dealing with lots of those kind of problems.

JB: Yeah, and we also did a lot of work with Los Alamos—

GB: Yes. That would have been of course post Manhattan Project kind of work, but there was a lot of nuclear testing going on and—

JB: Let's see. When was the Manhattan Project?

GB: The Manhattan Project was what, late-- The bomb exploded in '45, yeah.

Gardner Hendrie (cameraman): The bomb exploded in '45. The first hydrogen bomb was, I'm thinking '52.

GB: That sounds right to me. But then we were doing a lot of nuclear testing because we didn't really know the impact of-- We were shrinking them and Los Alamos was doing quite a bit of work there. In fact, to jump ahead, I do work with people in the supercomputing domain, and

Fortran still rules in that space—

JB: Yeah, I've heard that.

GB: Most of the super computing work is still all done in Fortran.

JB: That's astonishing.

GB: It is astonishing. But a lot of the weather forecasting programs…

JB: Well, they have all of those dusty decks to deal with.

GB: I don't think they use punch cards anymore.

JB: I know, but the equivalent thereof.

GB: The equivalent of. But it's amazing how much that software still lives, because they are probably using those algorithms which existed decades ago. In some of the nuclear simulations and weather simulations and stuff -- that stuff is still pretty much all Fortran these days.

JB: Uh huh. Wow. Astonishing.

GB: That was Fortran 1. What led to Fortran 2?

JB: Well, it was— [pause] You couldn't-- There weren't any subroutines in Fortran 1. You just-- It was one big mess. So Fortran 2 added the ability to have subroutines and—

GB: And the notion of subroutines coming from Maurice Wilkes I think—

JB: Yeah.

GB: --first proposed that notion. Did you have much interaction with his team at all?

JB: No, but we read about it.

GB: So subroutines were the big thing in Fortran 2. JB: Yes.

GB: What did that enable you to do? Was it hard to introduce the notions into the compiler?

JB: No. It was-- It didn't take much doing. I think Irv Ziller did most of the programming that made that change from 1 to 2. Irv Ziller and Libby Mitchell. She's absolutely disappeared from this earth. I've tried to locate her but I have not succeeded.

GB: I'd read she was kind of the program lead for Fortran 2.

JB: Huh?

GB: She was kind of the program lead, the project manager—

JB: No. Irv was. GB: Irv was. JB: Yeah.

GB: What was Libby's role in Fortran 2?

JB: She did a lot- just a lot of programming. She was a very good programmer and she did it accurately and quick.

GB: What begat Fortran 3? What were its novel things? I've read of Fortran 1, 2 and 4 but 3 seems to have disappeared.

JB: Well, 3 was mainly again Irv's thing, where you could sort of combine symbolic programming and Fortran programming.

GB: Fortran 4 is the one that really seemed to gain traction with the commercial world.

JB: Yeah.

GB: You were still deeply involved in that work at that time, weren't you? JB: I don't think so. I think that was mainly something that Bill Heising did. GB: By that time you were still at the Watson Labs. Is that correct?

JB: I was never at the Watson Labs. GB: You were just in New York City? JB: Yeah.

GB: Interesting, because I thought I'd read somewhere you were at the Watson Labs. You stayed in Manhattan this whole time then?

JB: Yes. Yeah.

GB: Interesting. It wasn't until later that you went out to the West Coast.

JB: No. Wait. Let me get this straight now. When you said Watson Labs I keep thinking of some little place on 110th Street but—

GB: I meant upstate New York a little bit.

JB: Yeah. You meant out in Yorktown. Yeah. Let's see. When did I move there?

GB: I couldn't find a date for that one because you were in New York City for the most of the '50s and then you ended up at the San Jose Labs I think in '63 so—

JB: Right. I wasn't at the Yorktown Lab for very long, but I can't remember when I moved.

GB: No worries. With Irv and others taking more up on Fortran 4, where did you find your work leading? Where was it leading you?

JB: Well, it's leading me nowhere basically because I had an obsession with the four-color problem. It was just some crazy ego thing that I was totally ill equipped to deal with, but I had some idea that I was going to generalize it and thereby make it easier to solve, or something like that. It was simply that the idea was to prove that any map on a plane could be colored

with four colors without having neighboring countries of the same color. I messed with that for ages and ages and ages and never got-- I had all these nutty little theorems that I proved, but never really did it.

GB: Although identifying yourself not as a mathematician, you were dealing with some pretty deep mathematical issues.

JB: Well, I was dealing with a difficult mathematical problem but I would never say that I was dealing with deep mathematical issues.

GB: And IBM gave you the freedom to pursue these things.

JB: Yes, they just left me to stew in my own juices.

GB: Did you have other colleagues you were working with around that time or pretty much a loner in that?

JB: I was pretty much a loner doing that, huddled in my office in Yorktown and later out- when I moved to the West Coast.

GB: Let me go back to the Fortran days for one more moment on that. As you look back on what you guys did there, what do you think you got the most right, and what do you regret having done, the most wrong, if you will?

JB: Well, I think history has shown that we divided up the problem pretty much right because a lot of people have kind of followed that same pattern in general.

GB: You really just stumbled into that, didn't you? The way the compiler and such was divided up? Or was it just sort of the natural decomposition that fell out?

JB: Well, it just-- We started to do it as one unit and then we saw that, uh oh, that isn't going to work, so we had to subdivide the problem into two things, and then that kind of went on like that. It was a pretty natural process.

GB: In all, you ended up with-- What was the final form of the compiler?

JB: It had six sections—

GB: Six sections.

JB: Yeah.

GB: It manifested itself as 250,000 cards, or some huge number like that? JB: No! The original compiler was, if I remember, about 30,000 instructions. GB: Wow. So efficiency and smallness were virtues at that time.

JB: Well, people were writing these things [instructions] one by one. You didn't have these programs that could slosh out thousands of instructions from one little piece of writing.

GB: Remarkable. Does the museum have a copy of that first compiler now? I think you were saying that they do, they don't?

Gardner Hendrie (cameraman): I believe they do have a copy.

GB: How exciting.

JB: Yeah. I think I've heard that they do have one of the early compilers, yeah.

GB: That's remarkable. After the Fortran work, did you have much contact with the Fortran communities that eventually standardized Fortran internationally?

JB: Well, my main contact was coming to some-- What was it called? The-- Some committee that dealt with that stuff.

GB: ISO or—

JB: No.

GB: ANSI? I forget. There were so many standards groups around that time.

JB: No, but this wasn't a standards group. This was just the committee within- all the aerospace people and stuff that dealt with Fortran.

GB: They wanted to get some common standards.

JB: Yeah, so that they'd have a uniform thing and they used to- that committee used to hold meetings in Los Angeles and various places which would mainly be getting together and-- Who was it? I guess it was usually Roy who would come-- No. It was a guy at North American, not Frank Wagner but another guy, a little guy with a moustache. Anyway—

GB: We'll do pattern matching with pictures here.

JB: Yeah. He used to show up with big bags of liquor and we would have our meeting, which was mainly drinking—

GB: From this the standard of Fortran was born. Hmm. It tells me something about the standards process. I was involved in the standards process with Ada and other things and we didn't have any such meetings.

JB: Oh.

GB: They were much less fun.

JB: Yeah. Well, that went on for quite a while. Then it got the bigger bosses in the process, they got annoyed with us and sort of got it much more sober and productive, but I didn't participate in that.

GB: How did you get drawn into the ALGOL process then? ALGOL was my second programming

language. I learned Fortran 4, and I went to the Air Force Academy where we had a Burroughs machine and we were taught ALGOL 60 as our primary language. I loved ALGOL. That was a great language. I enjoyed it. It sounds like you have other opinions of it—

JB: I'm glad you liked it—

GB: How did you get drawn into that process?

JB: Well, I don't know that I was very much drawn into it.

GB: Your BNF work came from that of course—

JB: Yeah.

GB: --when you were specifying the structure of the ALGOL.

JB: Yeah, but I-- It was a big committee deal,

GB: Sure.

JB: --and I don't think I contributed very much to it.

GB: How did you and Peter get together, Peter Naur, to specify the syntax?

JB: Well, I had come to one of these meetings with this BNF description. It was a little paper and I handed it- I hand carried it because it was so late in the game, so I had these copies that I dragged to the meeting and passed out to people and nobody paid any attention to it.

GB: Really.

JB: No. Except Peter Naur. When he came to write up the thing, he used this descriptive method, and improved it in the process.

GB: In retrospect, I can't imagine specifying a language without using BNF. So what were they doing before that time frame? It must have been—

JB: They were just writing English.

GB: And giving examples, and stuff like that. I had read you were influenced by some of the work of Noam Chomsky, that led you to that.

JB: Yeah, well, that's a funny story. That's what I said and what I believed, and yet… Who was it? Somebody sort of proved that I was wrong about it, that I hadn't got it from Noam Chomsky, because the dates were all wrong somehow. But-- God, who was that?

GB: No worries. What became the de facto way, really, to describe the syntax of languages was something that I heard you say just came in late in the game in ALGOL. Now we look back on it and I would have sworn I thought it was a fundamental piece of the creation of ALGOL but—

JB: No. They were inventing that language long before and the description was pretty much of a mess.

GB: But the BNF work, it's been so influential in other languages. I know ADA picked up on it. Didn't Wirth use it to describe PASCAL as well?

JB: I think so, yeah.

GB: And pretty much every language. Did you have any interaction with Niklaus Wirth at all in the PASCAL work?

JB: Not in designing PASCAL, no. But we would meet in these meetings quite often.

GB: In your design of Fortran, are there things that -- "regret" is the wrong word, but it's "I wish we had done this thing a different way because we now created generations of programmers who are doing these terrible things". Is there any such thing in Fortran like that?

JB: Not that I'm aware of, because I never used it very much so I—

GB: Really! Interesting. So as a programmer, what did you program in most of the time?

JB: I never wrote many programs. I was not good at doing that. What kind of programs was I going to write anyway?

GB: From the four color problem -- that was around the time of the work with ALGOL, if I'm not mistaken -- then what did it lead you to next? Was this around the time you went to the [IBM] San Jose Labs, Santa Teresa Labs?

JB: Yeah.

GB: What led you to be moved out that way? I love the west coast; it's not a bad place to be.

JB: Well, I got an invitation from Berkeley to be a visiting professor there. I went out and I spent a year at Berkeley.

GB: Still with IBM though.

JB: Still with IBM. Never gave a lecture, just sat around in the engineering building. At the end of the year they were so pleased with my performance they invited me to be a visiting professor for another year. Again, never gave a lecture, and so that's how I kind of got moved out to the West Coast.

GB: The labs were really just forming around that time in Santa Teresa, weren't they?

JB: Yeah, so I didn't show up in San Jose for quite a long time. I was just hanging out in San Francisco, and finally they said, "Well, how about showing up?" So I started going down there, and after that I got working on this stuff about—

GB: The functional programming work?

JB: Yeah.

GB: The Santa Teresa Labs are interesting; they're still kind of in the middle of nowhere. Back then they must have really been in the middle of nowhere, because San Jose hadn't quite expanded its borders that much. And you were there when there was still a lot of orchards around in the area.

JB: Yeah. When I first went down there, there was just this little-- There was the plant and then there was this little building—

GB: What were they doing at the plant? What did they manufacture at that time?

JB: Oh, I think printers or something. I'm not sure.

GB: That sounds about right. I think these days there's a lot of database work that goes on in that space.

JB: Yes.

GB: You probably had the feeling of "I wish I had bought lots of real estate when I first moved into that area", because you saw some amazing transformations in the valley over the years.

JB: Oh, yeah.

GB: This would have been in the early 1960s you ended up in San Jose?

JB: Let's see—

GB: I think I have the date. Sixty three was around the time you moved there—

JB: Uh huh.

GB: --and that was way before Silicon Valley came to be Silicon Valley.

JB: Right.

GB: It was really around that time frame that you started your first forays into the functional programming work.

JB: Yeah.

GB: Tell me a little bit about the work environment in that space. You showed up occasionally at the labs it sounds like.

JB: Yeah. I mainly stayed home. I worked with Ted Codd briefly, and—

GB: This is before Ted Codd of the database world.

JB: Yeah.

GB: This was long before he'd really begun to formulate his relational database work.

JB: Yes.

GB: What work were you doing with him?

JB: Just trying to work on this functional programming stuff. I was just groping around really, trying to—

GB: What led you to the functional programming domain in the first place?

JB: That's hard to say because-- I was just trying to think of some sort of really higher level programming that wasn't as difficult as Fortran. The problem was that the idea of functional programming, the "combining forms" and stuff like that, came pretty easily. But trying to make it into a real full system where you could deal with all the other issues that you couldn't express in that language got very confusing and messy.

GB: Indeed, as I look over your career, you probably spent more time on functional programming than you did on the Fortran work certainly.

JB: I think so, yeah.

GB: Were there others around you that influenced the early functional programming work, or was this sort of a foray on your own?

JB: I think it was mostly a foray on my own.

GB: And IBM created an environment that they let you go off and do these wild things. JB: Yeah. I guess-- By that time I was a Fellow so I could do whatever the hell I wanted. GB: You were one of the first Fellows. Were you in fact in the first batch of Fellows? JB: Yes.

GB: Who were some of your peers who were the other Fellows or—

JB: I didn't know them, because they were all into printers and hardware and stuff, so I didn't know any of them.

GB: As I recall, the Fellow program was started by Watson Jr--

JB: Yes.

GB: --as a means of recognizing some of IBM's top technical talent, and let them alone so they can create new great things for us as well.

JB: Yeah. The idea was that you could do what you wanted for a year and--

GB: It turned out to be longer than a year though.

JB: Yes, and then I kept sort of doing what I wanted and they kept saying "Oh, yes, go ahead, do it."

GB: Very nice.

JB: Yes. Then they finally decided that they'd better quit this one year at a time thing.

GB: Then turn it into: this is sort of what you do the rest of your career?

JB: Yeah, sort of. That is the understanding, isn't it?

GB: That's the understanding. As I said to you, Nick Donofrio basically described that I have two jobs: One is to invent the future, and the other is to destroy bureaucracy. Of which there is a little bit of that around the organization. But it sounds like you were pretty much unencumbered by bureaucracy.

JB: Yes, I was very unencumbered by bureaucracy.

GB: That must have been nice.

JB: Yeah.

GB: Were you having as much fun then as you expressed you had in the time you were working on the SSEC? Or was it a different kind of fun?

JB: Well, the most fun time was working on Fortran. We really had a ball working on that because it was just… We had this nice place to work, and we all got along very well with each other. At one point we were in this hotel across from Saks Fifth Avenue, and everybody spent half their time looking in to the dressing rooms across the street.

GB: There's a distraction.

JB: Yeah, it was a distraction.

GB: Do you have much contact at all with any of your colleagues from that time frame? They must have scattered to the winds.

JB: Yeah. No, I have contact with Irv Ziller.

GB: Where is Irv these days?

JB: He's in New York, the same old place he's lived his whole life, across the Hudson, just in Yonkers or someplace. Not Yonkers, it's Riverside, yeah.

GB: He stayed with IBM for a long time then.

JB: Yes, he did.

GB: IBM tended to hold on to people a long time.

JB: He became very closely associated with the vice president, some-- I forget the name of the vice president but he was sort of an executive without portfolio. He was a very smart guy.

GB: Lois and Roy and Pete and Bob and Dave, any of those other folks? JB: Well, a lot of them are dead. Peter is dead, Harlan is dead, Roy is dead. GB: Lois?

JB: Lois is still alive. GB: Time marches on. JB: Yeah.

GB: Let's move back to San Jose. Do you think you were a Fellow? That would have been '63, around the same time. Was that at the same time you were moved over to the West Coast?

JB: Yeah.

GB: So you began the functional programming work. Tell me about some of its earliest formulations. What were the problems you were trying to solve [those] that Fortran didn't quite solve?

JB: Well, it was just trying to be at a higher level so that you didn't have to get into all those gory details and stuff. Basically, the idea was to try to describe the transformation that you wanted to take place, rather than how to do it. It evolved very slowly and peculiarly. I had no idea about combining forms when I started, and just finally hit on that idea and developed it.

GB: It really fermented in a time frame where there was an explosion of other languages. There was PASCAL, there were the successors to ALGOL, there was what became Ada. What did you think of the whole Ada? You were around that time frame.

JB: Well, I thought it was a pile of stuff—

GB: Tell us what you really feel, John. [Laughs] Don't hold back here.

JB: Well, it was just so incredibly bureaucratized and complicated and impossible to learn.

GB: Your functional programming work was so different.

JB: Huh?

GB: Your functional programming work was going down such a different path, of pushing simplicity and power of expressiveness and the like.

JB: But it was ultimately unsuccessful because it couldn't take in all the peripheral stuff that you had.

GB: Your functional programming work was unsuccessful.

JB: Yes.

GB: Let's dwell upon that for a moment, because there were some other papers that I saw after your Turing Award lecture ["Can Programming Be Liberated From the von Neumann

Style", 1977] which was also a turning point. Because I'd describe it as a wake-up call to the language developers and programmers, saying there's a different way of looking at the world here. The world was going down a very different path. Let me pursue that point of why you think it didn't succeed.

JB: Well, because the fundamental paradigm did not include a way of dealing with real time. It was a way of saying how to transform this thing into that thing, but there was no element of time involved, and that was where it got hung up.

GB: That's a problem you wrestled with for literally years.

JB: Yeah, and unsuccessfully.

GB: After the Turing Award work --that was 1977 when you were given the Turing Award -- and that paper was incredibly influential. I know there was some other work in functional language at Berkeley, and others that really picked up on it well, that tried to make that more concrete.

JB: Right.

GB: How long then did you continue on with that -- really until your retirement in '91? Were there other problems?

JB: Yeah.

GB: That was also around the time of lots of other languages. We saw the beginnings of C, and what became C++, and Smalltalk was around that time. What did you think of those languages?

JB: I never learned them, so I don't have too much of an opinion.

GB: If I were to ask you what your favorite language is, it would probably be? If there is such a thing.

JB: Yeah, I don't really have one.

GB: What do you think of the contemporary languages such as Perl and Python and Ruby? Have you tracked any of those things that are going on?

JB: No. I'm a terribly unscholarly person, and lazy.

GB: Really!

JB: Yeah.

GB: Interesting. I would never have characterized you as such.

JB: Yeah, it's true.

GB: So Fortran was a way to help deal with that laziness in a way. You could do things better—

JB: Yeah. That was my motivating force in most of what I did, was how to avoid work.

GB: That's not a bad thing necessarily. You said one time that the assumptions under which Fortran were made simply don't exist anymore, those assumptions being we needed to build really efficient programs. What do you think are reasonable assumptions for a language designer today for them to have to worry about?

JB: Well, I guess the question of it still seems that programming is a pretty low-level enterprise, and that somebody ought to be thinking about how to make it higher; really higher level than it is.

GB: But functional programming wasn't a fruitful path, you think.

JB: Well, it was and it wasn't. It's just that that whole paradigm didn't include these other [things]. Somebody needs to find a way to include those other things in a clean way.

GB: Do you think it's still possible?

JB: I guess. I don't know. It's a difficult problem. GB: It is, and it's one that consumed you for a long time. JB: Yes.

GB: You continued on that path for the longest time at the San Jose Labs. Correct?

JB: Yeah.

GB: Until your retirement from IBM in 1991 or thereabouts.

JB: Uh huh.

GB: Why did you leave IBM in 1991?

JB: I don't know. It was my official retirement age so I just—

GB: So it was time to move on.

JB: Yeah.

GB: You would have been 60 something—

JB: -Five, yeah.

GB: Sixty five.

JB: Yeah.

GB: I didn't realize IBM had such an age. I don't worry about those things. That's long in the future for me I guess.

JB: Good.

GB: From there you went on to Berkeley for a while and stayed there? Or had you taught for a while?

JB: No. That was earlier.

GB: What did you do after retirement from IBM?

JB: What did I mainly do? I didn't do much of anything, actually.

GB: You were living in San Francisco at the time, which is a nice place to be living.

JB: Yeah. I did a lot of work helping my wife get her computer stuff.

GB: You were tech support for the household.

JB: Right.

GB: That was a time frame when there was such an explosion in activity in the Silicon Valley and the whole area. Do you have any impressions of that timeframe?

JB: No, because I wasn't involved, and I had sort of gotten out of technical stuff. I was interested more in music and reading, and stuff like that.

GB: That's right, you're quite a classical music fan, and an abstract art fan.

JB: Well, I don't know much about abstract art actually.

GB: Tell me about your love of music. Did you always have that love of music?

JB: Yeah. And I always tried to sort of get into contemporary popular music, but I never managed to like any of it because it always seemed so trite.

GB: Do you play an instrument yourself?

JB: No, I don't unfortunately. I always struggled when I was younger to play the piano, but I was always clumsy at it, so I never….

GB: Who are some of your favorite classical musicians? Are we talking classic, classic like Bach and Beethoven and the like?

JB: Yeah, and Mozart is one of my favorites but I also like this- a guy named Goretski. He's a modern Polish composer. He does a lot of nice stuff.

GB: Interesting. And you have a love for art, and a love for reading as well, I think you said.

JB: Yeah, I like to read.

GB: What do you like to read mostly?

JB: History, biography.

GB: What are you currently reading?

JB: I'm reading this thing by David McCullough, a biography of Theodore Roosevelt.

GB: If someone were to read a biography of John JB, it'd be a pretty thick book I imagine. What would you hope it would say?

JB: Well, that I helped. That I contributed to the development of computing. And I'm essentially nonviolent.

GB: You're a peaceful, gentle guy. As you look back over your career, what are you most proud of? Is it the Fortran work? Is it the functional programming work? Is it something entirely different?

JB: I guess the functional programming stuff. Yeah.

GB: What would be your advice to somebody that would want to take up the banner of functional programming? Where would you suggest they begin, and what hard problems would you like them to pursue?

JB: Well, trying to functionalize all the input/output stuff.

GB: Helping it talk to the real world.

JB: Yes.

GB: Do you think there's a class of problems for which functional programming is better suited to solving than contemporary languages?

JB: Well, any problem that just wants to transform some piece of data into another piece, I'd say functional programming is better suited to that.

GB: In fact, we were talking earlier that Google even has a functional programming language, a thing called Goopy if I'm not mistaken.

JB: Right.

GB: It's exciting to see them picking up on it. So see, your work has gotten traction. It truly has.

GB: [Let's go] way back before Fortran. Where were you born?

JB: I was born in Wilmington, Delaware, a pretty lousy place.

GB: How was it lousy?

JB: Well, it's the home of the DuPont family, so it's a whole bunch of rich sons of bitches acting up. Acting rich. Not a very good place.

GB: And that would have been... Give me the year, I can't calculate it in my head.

JB: I was born in 1924.

GB: 1924. Your parents -- a little bit about their history. What does your dad do?

JB: My father was originally a chemist, but then during the first World War, he was a munitions officer. When he came back to the DuPont company -- he was told they were going to hold his job -- and they hadn't. So he became a stockbroker, and got really rich. That's the story of my father. He was born in Virginia, [to] a very poor family, and only went to two years of college, or something like that. He was a smart guy. Not a very nice guy, but he was smart. My mother died when I was eight and a half. And, of course, until the late '80s, early '90s, I hadn't remembered anything about her until I took some LSD and remembered a lot of stuff that I would just as soon have forgotten, where she was sexually abusing me. But it was amazing. Until I took LSD it was as if she hadn't existed; I had really just wiped her out of my memory. I couldn't remember anything about her, except one incident in which she plugged in a light thing and it short circuited.

GB: The things one remembers. Wow.

JB: That's the only thing I remembered about her.

GB: Did you have siblings? Any brothers or sisters?

JB: Yeah. I have an older sister who's now dead, and I have a younger brother who lives in eastern Maryland. Very nice place.

GB: So you were the middle child?

JB: Yes.

GB: Were you a precocious child? Were you energetic, a sports kind of child?

JB: No. I was a withdrawn, I think, rather sadistic child.

GB: Sadistic in what way?

JB: Well, I used to get kids in this little shed in the back of our house, and sort of mess around with them in some nasty way, humiliating way.

GB: You went to school in Delaware?

JB: Yeah. I went to a school called Tower Hill School in Wilmington, and then in the eighth grade I went to the Hill School in Pottstown, Pennsylvania.

GB: Did your dad remarry at all?

JB: Yes. He remarried to a really horrible woman, my stepmother, who was really neurotic as hell.

For a while she was an alcoholic, and would sort of hang out the window yelling at trades people, and stuff like that. Then she quit that, but she was always a bitch.

GB: So you grew up in that area, and then after high school you left the region?

JB: Well, after I went away to Hill School, I practically never came back to Wilmington.

GB: So your family stayed there, but you then went away.

JB: Yes. After I got out of prep school, I moved to New York. Got this wonderful 18 dollar a month apartment.

GB: Oh my goodness. Wow. Too bad you can't do that these days. Eighteen dollars a month?

JB: Eighteen dollars.

GB: That might buy you a second these days, in New York City. So what led you to New York? Remind me of that story.

JB: I don't know. It seemed like a good place to get away from my family.

GB: Definitely not a bad place to be. In fact, as you think about being in grade school and high school and stuff, what did you imagine you wanted to be when you grew up? Or were you thinking of those kind of things?

JB: I didn't think about that at all. After I got to New York, my big ambition was to build a really good hi-fi set, which was not around in those days. I got this huge chassis, that had these gigantic transformers and stuff. Didn't work very well, I must say, because I wasn't an electrical engineer. But I tried.

GB: Part of the history of you just trying things and not knowing that you were going to fail. And just do it.

JB: Right.

GB: Fascinating. In fact, I saw this as a theme in one of the interviews you did. You were talking about the book, "In Search of Excellence," I remember, which you talked quite a bit about. I think you reflected upon the importance of failure, [and] what we can learn from all that. What have you learned from your failures, would you say? Both software and otherwise?

JB: It's a difficult question. I've learned not to be too optimistic. To realize that doing something worthwhile is tough. Things like that.

GB: In fact, at lunch we were talking about an aspect of that. One of the things that motivated you in your work in Fortran, and subsequently in functional programming, was this growing complexity. You used a phrase about complexity that I thought was interesting.

JB: Oh, the "cesspool of complexity".

GB: Yeah. Talk about that. What is this "cesspool of complexity?"

JB: Well, I mean, just look around at software, and you see it everywhere. You see the contents of the cesspool, so to speak. Everything is so complicated. Everything comes with a manual that thick, and it's a mess.

GB: How should we attack that complexity? I think Fred Brooks once spoke of it as the "inescapable complexity" that exists. Is it truly inescapable, or can we master it?

JB: Well, that remains to be seen. Actually that functional programming was an effort to try to go up a level, so that you didn't have to keep saying how to do everything, but rather say what you wanted done. That idea of saying what you want done bumps up against a lot of problems in input and output, and stuff like that.

GB: All those real things.

JB: But that's basically where it should go.

GB: That reminds me, speaking of Fred for a moment. Did you have much interaction with the 360 project and Fred's work?

JB: Well, I had some interaction with it. I was involved in one of the design meetings and stuff. I kind of dislike the whole idea of different machines. I was really pushing for an identical design for the three middle machines in the class, and I don't know whether that happened or not, really. So that programs would be all the same for them.

GB: Interesting. That was kind of a "bet the business" move on IBM's part, the whole 360 thing. It definitely transformed the company. Happily, that bet went well for them. Do you think there are any other big bets companies should be making these days?

JB: Well, yes. If they can really bet on software that will make it possible to say what you want done rather than how to do it. I think that would be nice.

GB: Here's a philosophical question, speaking of software. Is the world a better place because of all the software that's been written in your lifetime, or not?

JB: Well, in human terms, probably not. Because it just takes us further and further away from human affairs. But as far as economic, and welfare, it's done a lot of good. So it's a mixed bag.

GB: It is a mixed bag, it really is. You strike me as just an amazingly very human and gentle and caring person from that answer, and I absolutely love that. Where do we go from here? What do you think is going to happen in my lifetime?

JB: Well, I don't know. But I don't envy you, I'm afraid. I think that we're getting more and more technological and less and less human oriented. And as a country, we're getting tremendously aggressive, and we're going to pay for it. So it's a tough call.

GB: I'll sleep well tonight now. <laughs> I meet with a lot of people who are considering, should they pursue a career in software or not, hardware or software, anything in this space at all. Any advice you might offer? If you were to talk to somebody, say in high school or something like that, saying, "Gee, what should I do here?", what would you say to that person?

JB: Well, don't go into software. It's just such a complicated mess that you just frazzle your brains trying to do anything worthwhile.

GB: Understood. Looking back on it all, have you had fun?

JB: Yeah, I had a lot of fun.

GB: And the most fun you talked about was the Fortran days and the SSEC...

JB: Yeah, that was a lot of fun. I had a lot of fun too working on the functional programming stuff.

GB: That must have been, because you were breaking completely new ground there, and you were unfettered by legacy or anything like that to pursue it.

JB: And I had good, nice people to work with.

GB: Who were some of the contemporaries you mostly worked with in the [effort]?

JB: John Williams, primarily.

GB: Do you stay in touch with John? Is he still active in this space?

JB: Yeah.

GB: So many people you have interacted with over the years. Many of them were my heroes— you're one of my heroes— I'm delighted to have met you. I think I've run out of questions.

JB: Good.

GB: Anything else you would like to ask?

Gardner Hendrie (cameraman): No, but thank you.

GB: Thank you so much. This was a delight, to have a chance to meet you.

JB: I've enjoyed it.

GB: And I've learned some tricks about what it means to become a Fellow. So I'll institute them. Which says basically, "Just do it." The Nike approach. “Just do it.”

END OF INTERVIEW