Oral History of Robert Supnik, part 2

Interviewed by:
Gardner Hendrie

Recorded May 9, 2017
Carlisle, MA

CHM Reference number: X8215.2017

Hendrie: Let's see, I think where we left off. We were going to take a little detour and go through the history of the VAXes, the microprocessor VAXes.

Supnik: Okay. So as I mentioned last time, in 1979 when I joined what was then called the Small Systems Group, I spent the first six months as a systems analyst, looking at future strategy. As well as crawling all over the floor, checking the layout of the T-11.

Hendrie: <laughs>

Supnik: I recommended one last PDP-11, which became the J-11, and a VLSI VAX project, which became known as "Project Scorpio." The VLSI VAX project was intended to do a whole VAX. The thought of a subset never crossed our minds, back in 1979. Because the VAX architecture was the VAX architecture, you know, it was done by all the corporate luminaries, and who were we? We were just a bunch of kids and chip designers. So the Scorpio Project focused on the CAD tools needed to build a very big and complex design. Because it looked like it would take four different types of chips to build a VLSI VAX. A microcode chip, a microsequencer chip, a data-path chip, and a floating-point unit. I'm sorry, no. Microcode ROM and sequencer, memory management, data-path, floating-point.

Hendrie: Ah, yes.

Supnik: Okay. And so, that project spent the first couple of years really focusing on what I would call the foundations, of what kind of process we would need. Which led to the two-level metal idea, and what kind of CAD tools we would need.

Hendrie: Were there any CAD tools available…

Supnik: Very..

Hendrie: …at that point?

Supnik: …very few. There were some on the verification side, but there was nothing on the design schematic side. And nothing, above all, in the simulation arena. DEC was creating a new simulation system called "DECSIM," but it was still not ready. So the advanced development project created a schematic entry and librarian system, which was called "CHAS," C-H-A-S, named after the project leader's cat.

Hendrie: <laughs>

Supnik: Carol Peters was the project leader. And they struggled for a long time, we didn't have the right languages and tools and so forth. And when CHAS was finally ready, it didn't work very well. In fact..

Hendrie: Does that mean it was very slow?

Supnik: And very unfriendly. In fact..

Hendrie: <laughs>

Supnik: …they used to give out..

Hendrie: Could you define "unfriendly" in a piece of software? There are many definitions <laughs>.

Supnik: The users hated it.

Hendrie: Oh.

Supnik: That's what..

Hendrie: Okay, that's good enough.

Supnik: In fact, we used to give out foam pads to put on the top of your VT100, to protect your head when you started going <makes crashing noise>. (mimes banging one's head against the terminal)

Hendrie: Oh, oh.

<laughter>

Supnik: They were known as "CHAS protectors."

Hendrie: Okay. Now does Verilog exist at this time?

Supnik: No. Verilog did not exist.

Hendrie: Ah, okay.

Supnik: So eventually, one of the experienced designers that we had hired, named Ed Burdick, had enough of this. He redesigned the whole front end himself, based on his experience as an experienced chip designer, into a schematic entry editor called "Quickdraw." It ran on a VT100, amazingly enough, and was quite productive because he knew what he needed. And in fact, Ed over time moved out of chip design and into the CAD group, where having an actual designer resident and sort of helping with the ideas, really made the whole suite a lot more functional. So CHAS was eventually succeeded by a new one called "Katie," named after the new project leader's cat. And its motto was, "Katie, not user-surly." Not quite user-friendly, but no longer user-surly.

Hendrie: <laughs> Okay.

Supnik: So when Zilog came along and basically proposed doing a subset, and the big red light went on in our head, I started a parallel project, that drew on many, many of the components of the original VLSI VAX or Scorpio program. So the two were really running side by side, with Scorpio aiming more at a midrange system, and MicroVAX II, as it was known, being aimed more at a low-end, and actually a chip market. And then just to make things more complicated, once the MicroVAX subset was defined, Dave Cutler, who was trying to write a new software system for it called.. Micropower Pascal?

Hendrie: Okay.

Supnik: Or no, that was the PDP-11 version. But he was writing a small real-time operating system, which had never existed on the VAX. [He] decided that he would do a precursor system, out of a much simpler VLSI design methodology. He would just do the data path — the control ROM, the microsequencer would be external — [and it] would be much slower. And that ended up being called MicroVAX I.

Hendrie: I was going to ask, whether the two—- said it's the second.. Microprocessor VAX project. No.

Supnik: No. It's the first MicroVAX processor, but it's the second implementation of what was called the "MicroVAX instruction subset."

Hendrie: Okay, got it.

Supnik: You know, we managed to confuse ourselves as much as everyone else. MicroVAX I did ship about a year before MicroVAX II, at about a third the speed.

Hendrie: Mm-hmm.

Supnik: And [it] sold a respectable number, as a development machine. And also, it was the first VAX that got a graphics head and was sold as a workstation. So it was a very interesting system. Not too many of them have survived. So the difference in the level of effort we needed to do MicroVAX II versus Scorpio, which started much earlier and finished a bit later, convinced me and most other people that we really only wanted to build single-chip microprocessors from then on. And we only wanted to build to the subset of the VAX instruction set. So the second one was called CVAX, because it was our first CMOS VAX. It was aimed at being two and a half times the speed of MicroVAX II, [but] it turned out to be about three times. Very, very successful, shipped in... I want to say September of ’87.

Hendrie: That was shipped in systems too?

Supnik: Yes.

Hendrie: Okay, as well as…

Supnik: Right.

Hendrie: …other flavors.

Supnik: And both the MicroVAX II and the CVAX were used to create a line of workstations, called “VAX stations.” They were all designed by John Kirk, who had been the guru of the PDP-8 group. So John knew how to make things small, and he knew how to make them inexpensive. And the MicroVAX workstations were extremely successful in their heyday. In fact they were the second best-selling workstations in ’87 and ’88. Then the third design was called “Rigel.” So now we’re into one-and-a-half micron CMOS II. And it was basically a chip implementation of the big midrange machine called the “8800.” We just lifted the microarchitecture. The team had nothing but admiration for Bob Stewart, who had designed the 8800, and Doug Clark, who had been its principal architect. It was very successful, although we had a lot of trouble yielding the chips at first. It shipped, I want to say late ’88, or early ’89. It’s also the only VLSI VAX that ever had a vector unit on it. In the late ‘80s was when Convex and Alliant, and the mini-supercomputer craze hit. And so of course we had to have vectors too.

Hendrie: <laughs>

Supnik: So Tryggve Fossum of the big 9000 project, Dileep Bhandarkar from the VAX architecture group, and I figured out how we could shoehorn a vector extension [into] not only into the VAX instruction set, but into the Rigel implementation, which was already—- the first chip had already taped out.

Hendrie: Wow. Which chip taped out first though? The ROM?

Supnik: The cache.

Hendrie: Okay.

Supnik: The cache chip on the—- Rigel had a microprocessor, an external cache controller, and a floating-point unit.

Hendrie: Okay.

Supnik: And the really hard part was that we couldn't extend the microprocessor.We had very little ROM space left. But we did it. And then the midrange systems group designed the actual VAX vector chips. That was the only VLSI VAX that ever shipped with a vector unit. Didn't ship very many of them, but…

Hendrie: There wasn't—- yeah, you did not really successfully compete with Alliant and..

Supnik: No.

Hendrie: …Convex.

Supnik: And they were flashes in the pan anyway. RISC machines wiped them out pretty quickly.

Hendrie: Yes.

Supnik: And then the fourth—- I'm sorry, the fourth VLSI VAX was named "Mariah." And it was simply a shrink of Rigel. Rigel had been done in a one-and-a-half-micron process, and Mariah was the same chipset moved to one micron, with an appropriate boost of speed. And the cache was changed from write-through to write-back, to improve performance. So it was very successful as sort of a midlife kicker. And then the last VAX that we designed was called NVAX. Done in CMOS IV, which was three quarters of a micron. And it was basically a full implementation, on a single chip, of the VAX 9000.

Hendrie: Wow.

Supnik: It had full instruction pipelining, [and] it had a built-in floating-point unit that was fully pipelined. It was much, much faster than the previous chips. And when it came out at the end of '91, it was the fastest CISC microprocessor in the world. Not only in frequency, but in actual performance. It just blew away everything that Intel had at the time. And that was then shrunk one more time, into the CMOS V process, which sped it up some more. And gave us the chip that we rode out the rest of the VAX lifetime with. So there were one, two, three, four, five designs. Scorpio, MicroVAX II, Rigel—- I'm sorry, Scorpio, MicroVAX II, CVAX, Rigel and NVAX. But there were many more implementations. CVAX got shrunk twice, first from two micron to one-and-a-half micron, and then from one-and-a-half microns to one micron. And both the external cache and the floating-point unit were merged in at that point. So you had a single-chip VAX, very small, very low power, for sort of low-end applications. Rigel got shrunk, NVAX got shrunk.

Hendrie: So as the processes went to, you know, as the lithography got better, fundamentally, and you got…

Supnik: Right, we got another half generation out of it..

Hendrie: Yeah, you would just…

Supnik: …to bridge the two-and-a-half… two to two-and-a-half years between new designs.

Hendrie: Yes.

Supnik: And it was a very, very successful strategy, in the sense of producing very competitive chips in systems. But because it was not in support of a volume strategy, we were using an exceedingly expensive design and manufacturing methodology to produce a relatively small number of chips. And the economics were not, from DEC's point of view, sustainable in the long-term. Because the foundry idea had not taken off yet. Foundries didn't really become a reality until the mid to late nineties. At least foundries with competitive processes.

Hendrie: Yes.

Supnik: So we were...

Hendrie: So you could have—- you probably could have gone to Taiwan for a one generation older process, at that time. But that isn't the thinking you're trying to..

Supnik: Well yeah, and we also wanted a process that had been tuned for speed, not for yield.

Hendrie: Yes.

Supnik: So it was all—- you know, we were very much wedded to the notion that the advantage we had as a design group came from the integration of design, CAD, process design and manufacturing. It was all on one side, it was all one team, and we all…

Hendrie: You actually talked to each other <laughs>.

Supnik: And we all worked to a common goal.

Hendrie: Yeah.

Supnik: And what was very interesting, the CAD suite really evolved over that decade, to incorporate more and more of the designers' wisdom into how it worked. And in particular, a tool was created, whose name I've forgotten, I'm sorry. But it was basically an encapsulation of what I would call "designers' wisdom." And after you had finished the design verification, proving that the schematics and the layout were the same, you then ran this design critic. And it looked for all kinds of things that no other CAD tool at the time could find. Cross-layer interference and crosstalk, variable and improper capacitive coupling. It was the work of Ed Burdick and another designer named Bill Grundmann, who moved into CAD. And they really, really helped to get the designs to be much more stable. So you remember [on] MicroVAX, we couldn't get it to run for two passes. [In] CVAX, there was a fatal bug in the first pass that prevented it from running VMS. Rigel ran VMS on the first pass, and NVAX required only two passes to actually ship in systems. So, you know, I have an analysis of our bug rates, what the bugs were. As the bug rates declined, the kinds of bugs we found became sort of more and more on the fringes. It really shortened the time to build it, and it was all as a result of CAD tools.

Hendrie: Yes, okay.

Supnik: So that's the story of the VLSI VAXes. And sort of after Alpha shipped, there weren't any more VAX designs, there were just Alpha designs.

Hendrie: Yeah.

Supnik: So it's now 1992, we've just shipped Alpha, and all the people involved go on to reap their rewards. Bob Palmer, the head of the semiconductor group, is named the new CEO of DEC. Bill Strecker becomes VP of engineering. This is the first VP of engineering that DEC has had since Gordon left. And I get named Chief of Technology Strategy, otherwise known as Bill's assistant, right-hand person, gofer, I don't know.

Hendrie: <laughs> Okay.

Supnik: But it was now time to deal with the consequences of the Alpha business plan. As you may remember from last time, I said that if Alpha succeeded, it still required the company to shrink by 25,000 people. Because we assumed that the gross margins in Alpha systems, could not be sustained at the level of a VAX. Because we wanted to sell more Unix systems, and even after we made the alliance with Microsoft, some Windows NT Systems. And they would not command 60, 70 percent gross margins. Just wouldn't. So we were pursuing licensing the chip, we partnered with Mitsubishi there, and eventually with I think Samsung. And then we were looking at the structure of engineering. So fundamentally, my job in '93, as my reward for Alpha, was to figure out how to downsize DEC Engineering by about a third.

Hendrie: Whoa.

Supnik: Yeah.

Hendrie: Yeah. <laughs> That was not going to make you popular, besides being a very difficult task.

Supnik: Yeah. But surprisingly enough, because we had been so straightforward about Alpha, we had said, "This is what it is. And by the way, these are the consequences." I mean we didn't hide the business plan. People in engineering knew that we were going to have to make some drastic changes. In particular, we couldn't afford so many processor groups. We had to move more resources towards software. We couldn't afford to do proprietary designs in areas we had traditionally done them. And so rather than you're just going trim, trim, trim around the edges of everyone, the strategy that I recommended and we implemented was to close down entire functions that no longer made sense. So for example, we basically stopped designing power supplies. By this point, when you're building systems that are entirely based on microprocessor technology, even if you're scaling them up very big, there's a wide variety of industry power supplies that you can use. Some very reliable, all of them much more cost effective than anything DEC could design, because of our low volume. So we basically said, "We are going to stop designing power supplies." I think the people who did that for a living weren't thrilled, but they could understand the rationale. We said, "We can't have as many processor groups." We used to have a low-end group, a low-midrange group, a midrange group, a high-end group, and a workstation group. And said, "We can't do that anymore. We're going to have a low-end group, a midrange group, and a workstation group. That's it."

Hendrie: Okay. And all these processors groups were essentially designing to the same architecture, it wasn't like you have PDP15 group, and a...

Supnik: Right.

Hendrie: …PDP8 group, and a..

Supnik: They were all designing VAXes.

Hendrie: …PDP-11 group. Yes.

Supnik: And again, you know, they weren't happy when basically the high-end group was put out of business, by NVAX and then by Alpha. That was the end of a very long and proud tradition of design in Marlborough, dating back to the '60s. But they could see what the technology had done to us, and to them. A few of them chose to join the VLSI team, as architects; the rest decided to pursue other ventures. So I would say it was relatively amicable, as far as these things can be. We simplified the packaging drastically, we stopped doing these elaborate packages. We stopped doing any proprietary buses. We went over to industry-standard buses, and we stopped making proprietary disks. So we'll use SCSI drives, IDE drives on the low end, we'll use PCI as our standard interconnect. Which had the amazing effect of giving us common IO cards across the entire line, from the first time in DEC's history.

Hendrie: Really? Wow.

Supnik: And so these efficiencies interlocked with each other. In a way, that permitted us to take down the hardware investment, without in fact taking down the range of systems we could produce. Permitted us to put more into software, and permitted us also to produce more competitive systems and more cost effective systems. And in the meantime, DEC was shuttering manufacturing plants and closing facilities and so on. And Alpha was sort of trying to find its marketplace. And then in 1993 and '94, two things happened in parallel. First, the internet really exploded, the web exploded, and Alpha was the best web engine in the world. Because it was 64 bits. And it could have a lot more memory and virtual memory, it had really robust Unix. And then secondly, Oracle took a gamble and ported their database, not just to Alpha, but to the 64-bit version of Alpha. And their performance went through the ceiling. Because they could keep all, you know, much bigger sections of tables, at least in virtual memory if not in physical memory.

Hendrie: Right.

Supnik: And they weren't doing what was effectively software paging anymore. So suddenly we were Larry Ellison's best friend.

Hendrie: <laughs>

Supnik: Because we just killed everybody using Oracle for the transaction processing benchmarks. And then third, Alpha was the machine of choice for high-performance technical computing. It was just the fastest thing you could get your hands on, at the price. One of the most poignant stories I remember from Alpha, is early in its history, it's probably 1993. I went to EPTH… I think that's what it was, it's a polytechnical college in Switzerland.

Hendrie: Okay.

Supnik: And they had just bought a relatively big Alpha workstation, I mean it was about… the size of a full tower instead of a mini tower. And I went there, and we proudly presented it to them. They decommissioned a Cray 2 because of it.

Hendrie: Wow.

Supnik: So here was this workstation, costing I think less than 30 thousand dollars. And it had displaced a multimillion-dollar supercomputer.

Hendrie: Yeah.

Supnik: I mean... that was very impactful to see. Because… the engineering of all Cray machines was just marvelous, if you admire these things.

Hendrie: Yes.

Supnik: And that one was water-cooled, and I think it was one of the ones that had the transparent cooling tank, with the modules inside.

Hendrie: Yeah, I think so.

Supnik: And as usual, they had thrown some plastic fishes in, to bubble up and down in the Freon currents. It was lovely.

Hendrie: <laughs>

Supnik: And it was being de—- it had padded seats, because it would…

Hendrie: Oh yeah, all the way around.

Supnik: And it was being decommissioned for this beige box that we had built.

Hendrie: Well, that's what happened to discrete component machines.

Supnik: Yep. That was the living fulfillment of Gordon's graph. Microprocessors had crossed the supercomputer line.

Hendrie: Yeah.

Supnik: So as Alpha found its markets, the business began to grow. And DEC turned profitable… in '94, after having…

Hendrie: Had it had losses before?

Supnik: Oh, it had been losing money steadily since about 1990.

Hendrie: Okay.

Supnik: It was in the black, between having reduced the size of the workforce, and having very competitive machines. In '94, the second generation of Alpha systems came out, all standardized around common IO, PCI IO. All standardized around common storage, around industry-standard storage. And with now the Oracle results to drive commercial, the Internet results, and… <laughs> the Internet proof point was very interesting. So it's 1994, and *Sports Illustrated* decides that they are going to, for the first time, put the swimsuit issue on the web.

Hendrie: <laughs>

Supnik: Which they do. And their Sun SPARC web servers promptly crash under the load. And there's nothing they can do, can't put more on, they can't—- nothing seems to work. They call us in desperation, can we do it? We bring Alpha in. We're handling millions of page hits a day, all from one machine.

Hendrie: From one machine.

Supnik: From one machine. So that was also the year that I think Brian Reid's team, out at the Network Systems Lab, got involved with elections and posting elections on the web. It was local elections. Again, using Alpha. So Alpha became the go-to machine for the high-end part of the Internet. And it was the go-to machine for high-end transaction processing, the go-to machine for high-performance technical computing. It seemed like we had a winner, finally. Unfortunately, the company's management didn't have the same feelings. And…

Hendrie: Now was this now—- this, we're in the Palmer era.

Supnik: We're in the Palmer era.

Hendrie: Yeah.

Supnik: And when Bob was the boss of the semiconductor group, you know, I worked with him, I respected him, he gave me the best investment advice I ever got in my life. It was one word and he said, "Diversify."

Hendrie: <laughs>

Supnik: Because I was holding all my—- basically all my life savings in DEC options, he said, "Diversify." He gave me this advice two months before the October, '87 crash. Good advice.

Hendrie: Were you fast or slow?

Supnik: I was… fast.

Hendrie: <laughs> Okay.

Supnik: But cautious, I took…

Hendrie: Yeah, you don't do it all.

Supnik: …half of them, then the crash occurred. And then I sold the rest of them the following year, very early. And I made out the same, because between '87 and '88 was the Reagan tax cut.

Hendrie: Ah, yes, okay. All right.

Supnik: Anyway. As I said, Bob was a wonderful semiconductor guy, he knew operations to his fingertips, he knew all about manufacturing plant. He was a fairly indifferent judge of people. And the people…

Hendrie: Ooh.

Supnik: …he hired as business managers… were by and large not successful. And so he kept changing and changing structures, and finally he decided that the central structure of engineering should be replaced by business units. And he gave control of the business unit that included Alpha, to a fellow by the name of Enrico Pessatori,. who came out of the Olivetti PC Group, and who really believed in PCs.

Hendrie: Okay.

Supnik: So the PC group…

Hendrie: Oh yes, all right.

Supnik: PC group was given a lot more money, became a lot bigger, and the Alpha group was systematically sort of pushed to the side. And the company began to lose money again. In '96—- well let me—- so I was made Enrico's chief of strategy and technology. But I had nothing to do, because he wasn't interested in the Alpha side of it. And the PC side of it was, "Intel told me to do this, I'll do this." So I really didn't have any work to do. In '94 and '95, I took the longest vacation I had ever taken in my life, you know, because I'd accumulated, I don't know…

Hendrie: As the…

Supnik: …for 20 years I had been accumulating vacation, and DEC didn't have limits at that time. So I took like an entire summer off. And I started interviewing.

Hendrie: Yeah.

Supnik: And I was again interviewing on the west coast, because my children were out there for the moment, as it turns out, my son had moved to San Diego. And so I was interviewing on the west coast again, and then Bill Strecker, who was now chief technology officer… called me up and said, “How would you like to run the Research Group?” Well, the Research Grouop at DEC was a very, very concentrated group of extremely brilliant people. There was a lab under the renowned Bob Taylor, called “The Systems Research Center.” There was another lab called “The Western Research Lab,” under... who was running it at the time? Not sure I can remember. And then there was the Network Systems Lab, under Brian Reid. All of them in Palo Alto. You might wonder why we had three labs in Palo Alto. I wondered that too.

Gardner: <laughs>

Supnik: The Cambridge Research Lab in Cambridge, Massachusetts, which had been founded by Vic Vissotski, though he was no longer there, and we had a research lab in Paris as well, and this group produced brilliant, brilliant results, none of which ever made it into products.

Gardner: I was going to say, yeah.

Supnik: Right. It’s like, it was like Xerox PARC, you know. They were inventing the future and other people used it. The Systems Research Center did pioneering work on distributed systems that Sun realized in practice and DEC never did. So Bill, knowing that I had had a background in advanced development and product development, and had never worked in research, said, “I want you to get the labs to be more focused on shortening the span between their idea generation and product realization.”

Gardner: Yes.

Supnik: Fair enough. Now, I felt very, very uncomfortable stepping into the distinguished shoes of Sam Fuller, who had founded the research group or at least taken it over very early.

Gardner: Is that what he did most of his career?

Supnik: Yes.

Gardner: Yeah, okay.

Supnik: The Research Group was founded, I think, by-- now I can't recall the name, but Sam took it over in the early '80s and he built it to what it became. He recruited Bob Taylor. He recruited Brian Reid. You know, he recruited Chuck Thacker and--

Gardner: Okay.

Supnik: --Butler Lampson.

Gardner: Oh. So all-- yeah. A whole bunch of people from--

Supnik: Right. And--

Gardner: Yeah. From--

Supnik: And Sam was a--

Gardner: --Xerox, yeah.

Supnik: --distinguished computer scientist, with a Ph.D. He was Dr. Fuller, and I was Bob the math and history major.

Gardner: <laughs> Okay.

Supnik: So it was daunting to me.

Gardner: Now, had Sam left?

Supnik: No. Sam picked up the piece of the pie that dealt with universities and industrial consortia and some of the other functions.

Gardner: Yeah. Okay.

Supnik: He did leave somewhat later.

Gardner: He didn't leave for Analog until later.

Supnik: Yeah. A little later.

Gardner: Yeah, okay.

Supnik: So I knew all the people in research. I had been attending research reviews for a decade, you know, and I respected them as individuals but thought of them as, you know, I thought of them as a bit isolated from reality.

Gardner: <laughs>

Supnik: And so--

Gardner: They didn't, none of them had a drive to see it, you know, to take their ideas and see them used.

Supnik: No. I wouldn't say that. I would say they very much wanted to see their ideas used. They just didn't know how to do it. How the product development system worked.

Gardner: Okay. They didn't understand the-- yes.

Supnik: Right. To them it was, "Well, I wrote this paper, I made this prototype. You go run with it," and, of course, the problem is you give an engineer something like that, they're going to redesign it and rethink it and they're--

Gardner: Well, it isn't theirs.

Supnik: Yeah, the development time will stretch out.

Gardner: Yeah.

Supnik: So what I did was to first of all try and get the labs to focus on themes as opposed to whatever researcher, whatever a researcher wants to do, and our themes were the Internet, mobile devices, semiconductor architecture and-- I think the fourth was high-performance computing. It wasn't a mandate. It was, you know, if you're thinking about your next project, either find something in one of these four areas, you know, it'd probably be helpful. But things that were completely off the beaten path were allowed to go on. The research work that had been going on for close to 20 years in formal verification continued and then actually yielded results while I was there, strictly coincidentally. I had nothing to do with that. But what I did was to form an intermediate team. I formed an advanced development team under an experienced project leader named Mark Foster. Mark came out of the PC group, and Mark was the man who invented what we think of as the lightweight laptop. So in 1994, DEC had stunned the world with something called the HiNote Ultra, the first sub-four-pound laptop, thin and light. Just an amazing piece of design work. It was all Mark's, Mark's brainchild. But he was a pretty, pretty interesting individual. He was bright, he was driven, and he didn't fit so well in with the highly structured PC group that his bosses were envisioning, Mr. Pesatori.

Gardner: Okay.

Supnik: So he was happy enough to come work with me and the people on the West Coast, and a few of the developers, of the researchers, decided they wanted to be part of it too. It was never bigger than six or seven people, and the model we used was what had worked with AltaVista. You know, I didn't invent AltaVista. I wasn't there when it was invented. It was invented in '95 by Mike Burrows and Luiz-- forgot his name. [Louis Monier] Mike Burrows wrote the search, the core of the search algorithm, and the other fellow {Louis Monier] did the first true Internet scale spider.

Gardner: Okay. You mean the crawler?

Supnik: The crawler.

Gardner: Okay, yes.

Supnik: We called it a spider back then. I don't know why.

Gardner: Okay. Well, yeah.

Supnik: It crawled away. <laughs>

Gardner: I don't know. Yeah, yeah. I just know those things as crawlers. I don't know what they should be called.

Supnik: Yeah. And Alpha was key to both of them. Alpha, with its 64-bit capability and therefore the ability to map huge swaths of the internet in a single virtual memory space, made the crawler feasible and the search algorithm, what-- Mike was a genius, Mike Burrows. He's at Google now, of course.

Gardner: Of course.

<laughter>

Supnik: To get the search engine to be really fast, he took the core algorithm, and he hand coded it in assembly language. He hand scheduled it for the second generation Alpha chip, EV5's four-way issue, and he pared it until the core fit in the 8K byte or 2K word primary instruction cache. The reason that Alpha, this, AltaVista, was so fast is that the search engine was fundamentally running at 350 to 500 megahertz when everybody else was stuck operating out of main memory. Mike Burrows was a genius.

Gardner: That is fascinating. You know, I was involved in Wellfleet, which was doing comm work, of course.

Supnik: Mm-hm.

Gardner: And they, they did the same thing with the Motorola 68000, and it may have been the 68020. I don't remember exactly what model at that point, and that's how they were able to win, beat Cisco's router, which used a bit-slice, 2900 bit-slices, which were pretty fast and did the same thing. Guy handcrafted the basic loop until it--

Supnik: Until it fit.

Gardner: --fit. Exactly.

Supnik: And BBN did--

Gardner: It's a brilliant strategy.

Supnik: BBN did the same thing with an EV5, and as a result they had the first wire speed hundred megabit router.

Gardner: Is that right?

Supnik: Yep.

Gardner: Okay.

Supnik: They put that core loop in the EV5 primary instruction cache.

Gardner: Okay. Okay.

Supnik: Yeah.

Gardner: Yeah, that's… Yeah.

<laughter>

Gardner: It's a good idea.

Supnik: It's a good idea.

Gardner: So roll back just a little bit on AltaVista. You said that there was a-- that structurally somehow within DEC, I got the impression, that structurally the way it came about in DEC, it had done something, you know. It was set up in an unusual way--

Supnik: Very unusual.

Gardner: --so it didn't get trapped in the--

Supnik: Right.

Gardner: In all the process.

Supnik: So what happened was that after Mike and his collaborator [Louis Monier], and I should look up the name, because he was very bright too, had demonstrated it was feasible. Sam fundamentally said, "We're going to set up an AltaVista search service for the universe, and we're going to run it out of research."

Gardner: <laughs> Okay.

Supnik: So he persuaded Bill to find the capital to buy what was at the time a fairly significant number of big EV5 systems. So this is second-generation Alpha systems. We, he, reconditioned the first-floor data center of the western-- of the Network Systems Lab building, which was this old, old building sort of three streets back in Palo Alto, that just happened to be sitting over a fiber nexus.

Gardner: Okay.

Supnik: And he hired Barry Rubinson, who you may remember from other parts of my story as the guy who got me into hardware engineering, to run it operationally. So you had this complete 24/7 IT data center being run out of research with no commercial constraints other than be the number one search engine on the Internet, period, and it was.

Gardner: Yeah.

Supnik: It was the only one for a while.

Gardner: I remember.

Supnik: But, I mean, it was--

Gardner: Well--

Supnik: It was the one everybody went to.

Gardner: Lots of people were wannabes. I mean.

Supnik: Wannabes.

Gardner: But, you know, if you tried any of them, soon as you tried AltaVista you didn't go back.

Supnik: Right. So when I became the head of research, I really didn't have to do anything about AltaVista. It had the structure I wanted. It was-- I would've wanted, which is it was running almost as an independent little business unit. Wasn't clear how we were making money from t, but that wasn't really the issue at this point. It was, the view was, this is the best possible publicity to drive Alpha systems as the premiere Internet systems on the planet.

Gardner: Mm-hm.

Supnik: And so that's what we did. Kept buying more and more machines and plugging them into that space. Fortunately, Alphas got more powerful without getting much bigger.

<laughter>

Supnik: And we continued that strategy until Compaq enters the story in another year or two. So now with this focus on various areas, some new ideas began to come out of the woodwork. A gentleman named Bill Laing, who had been one of the most brilliant VMS designers, he had put symmetric multiprocessing in VMS, had lost his group in the U.K. It had been closed down, and so he had joined the research group as a remote researcher. He lived in Scotland. He was part of the lab in Palo Alto. He had a very long commute.

Gardner: <laughs>

Supnik: So once a month he would get on the plane, fly 14 hours, stay for a week and go home and then work from home. Well, his complaint was, at that time, the entertainment systems on airplanes were terrible. You know, they had common videotape loop. They might have these prerecorded music things that ran for an hour. His laptop wouldn't hold a charge, and there was no power -- no power outlets in those days, and he wanted music. He wanted enough music for him to last the trip.

Gardner: <laughs>

Supnik: Now MP3 as a format had just started emerging. There were these little Flash-based players that stored half an hour worth of music. So he [Bill] came and posed the problem to his colleagues, "Can you make a music player that will store enough of my collection to be interesting, will last long enough to survive this plane flight or at least have some way of changing the battery so that it can?" Out of that was born what was called the Portable Jukebox, and this was a little handheld device. Andrew Birrell, the late Andrew Birrell, unfortunately, did much of the work on the conceptualization of the UI, but the product design, the prototype design, was all done by Mark Foster's Advanced Development group. And because Mark knew, and knows till today, miniaturization to his fingertips, he figured out all the things that you had to do to make it small enough to be bearable. I mean, because it was going to have a hard drive in it if it was going to hold enough music, and yet it couldn't be much bigger than the hard drive itself. At that point, two and a half [inch drives] were the smallest we could get. It had to have decent battery life, had to have great sound, et cetera, and Mark, Mark and his team, with Andrew and some of the other researchers, they cracked this problem, totally. They figured out that you could actually power manage a laptop hard drive so that it was on very little of the time. You read a couple of minutes' worth of sound into a RAM, then you powered the laptop drive down. We didn't know at the time, and in fact that hard drive manufacturers weren't sure that you could power cycle a drive that often.

Gardner: Yeah.

Supnik: But it turns out it worked.

Gardner: Mm-hm. Yeah. Because of the head landing issues and…

Supnik: Yeah.

Gardner: Okay.

Supnik: So there were questions about how you program it legally at this point. All these little MP3 players were being sued by the RIAA. There was a question of, you know, there were just-- it was a lot of unsolved problems for that day and age, but they solved it. They solved them. That was one project that got started. I'll come back to the fate of the PJB.

Gardner: Yeah.

Supnik: There was a, there were-- there was a technology developed at the Western Research Lab by-- oh, my goodness. Alan Eustace, who is now a very senior VP at Google and-- I'm going to get his name wrong. I think it was Anant Agarwal but who's now a-- no. I know that's not his name. [It was Amitabh Srivastava.] But he, this fellow, was now one of the most senior researchers at Microsoft. All of this stuff I can get because the papers are still online. They had created a technology called Atom/Om. It started as a way of instrumenting binary programs so that you could answer performance questions, to which they added the ability to then rewrite the program based on what you learned from profiling. Binary to binary [translation] to improve its performance, and it really worked. It improved our Oracle TPC results by 20 percent. Oracle used it. VMS used it for its relational [database]-- everybody used it. So we decided to try another experiment. We would spin this out as an independent company to make a software product for the X86. We got some venture capital money and started that up. Another idea that came up was this notion of microcommerce, the idea that you should be able to have transactions on the internet at the fine grain level that you have on your cell phone. You know, I made this text, that call.

Gardner: Yeah.

Supnik: So one of the teams worked on that and actually came up with a practical microcommerce system called MilliCent, and we started trying to do some productization around that. The guys at the Western Research Lab, Bill Hamburgen in particular, who's their packaging guru, were extremely impressed with what Dan Dobberpuhl and Rich Witek had gotten up to after Alpha. They had made an ARM chip that was, like, 10 times faster than any ARM chip ever built, with not much more power. It was called StrongARM.

Gardner: Ahh, okay.

Supnik: And it was, you know, yet another brilliant achievement from Rich and Dan.

Gardner: Yeah.

Supnik: And they [the Western Research Lab team] realized that they could therefore build a handheld device that was far more powerful than the sort of random organizers that were available at the time, and use it to figure out exotic interfaces, touch, voice, positioning and so forth. It was called Itsy, and it was about this big. [About the size of a large iPhone, but thicker.] It looked like a personal organizer but it was a full-scale Linux machine at the time, and it had a touch screen and it had ports you could add hardware on, so you could put accelerometers and other things on it. Many of the things that would later come out in smartphones were pioneered there. Again, that, a common thread, is going to be that DEC was not very good at figuring out what business problem we were solving with our technology, but it was another wonderful, wonderful project. The Cambridge people did some wearable computers. The Systems Research Center chip designers or chip researchers said, "You know, instead of trying to build bigger and bigger and wider and wider Alphas, it would make a lot more sense to take a whole bunch of the last generation chips and put them on one die." So instead of building what would've been EV-8, which was never finished, because it was too complicated, which was an 8-way issue machine, they said, "Use the same process and put 8 EV-5 dies down on it. It'll take you a tenth the effort, it'll be even faster, and it's perfectly suited to both high-performance technical computing and commercial problems," which are intrinsically multi-threaded. So long before anybody was thinking about multiple cores, they had a concrete proposal for this.

Gardner: Wow.

Supnik: It was a brilliantly productive era, and yet the formal research, the esoteric research, continued as well. As I said, the formal verification, which Les Lamport had been working on for many, many years and which always looked like it was going to be kind of out there, they got together with the semiconductor group, and they did a formal verification of the cache protocols in EV6.

Gardner: Mm-hm.

Supnik: And they found an error. Fortunately, they found it before the design was complete, which was the whole idea.

Gardner: Yes. <laughs>

Supnik: So and he was--

Gardner: No one had to spin the chip.

Supnik: And it was such an obscure case. There was just no way in hell a human being would've found it.

Gardner: Yeah. Okay.

Supnik: So it began to pay dividends. But at this, by this point, DEC management had fundamentally given up. Bob Palmer had given up. He could not see how to save the company, and so he began to tidy it up for sale. I didn't realize this. I didn't realize this what he was doing, but--

Gardner: Mm-hm. He didn't tell people.

Supnik: --he sold the storage group off. He sold the networking group off, and then that left the semiconductor group and above all the semiconductor fab, an enormous capital sink if we were to stay current, and so a fairly clever and subtle strategy was designed. It wasn't necessarily an attractive object in and of itself, but if we got the attention of certain companies, maybe they would be interested. So we sued Intel for patent infringement.

Gardner: Okay.

Supnik: And I was part of the team that did the patent examination and helped select [the patents], and we really felt we had them dead to rights because they had been trying to make faster and faster microprocessors. They were pretty much following in the footsteps of Alpha. They even had some people from DEC, and they were using techniques we had patented. We pretty much felt we had them. So we sued them, and there were terrible consequences. Our PC group was cut off from all information and so forth.

Gardner: Yeah. Couldn't find anything about the new chips until they were publicly announced.

Supnik: Couldn't find-- couldn't get any confidential information.

Gardner: Right.

Supnik: But I was really looking forward to our day in court, and then a settlement was announced and the settlement did not include just a patent settlement. It included selling the fab to Intel.

Gardner: Yeah.

Supnik: And adopting the Itanium in place of Alpha.

Gardner: <laughs> Oh.

Supnik: Well, you can imagine--

Gardner: So where was Alpha going to go?

Supnik: It was going to go away.

Gardner: Oh, my God.

Supnik: You can imagine the shock inside the company and the sense of betrayal.

Gardner: Right.

Supnik: And I'm not sure that would've mattered, but the customers rose up and just said, "You do this, you're toast."

Gardner: Yeah. We're gone. We're gone.

Supnik: We're gone. So that part of the deal did get rescinded, but only that part. The fab was sold, and it was very interesting when the patent truce was announced. Intel then came back and said, "Yeah, okay. There's another 30 patents we'd like to make sure that we are clear on."

Gardner: Yeah.

Supnik: So yeah. They were coerced into buying the fab, I would say. That is just my opinion, you know. I wasn't there. I wasn't part of the negotiations, but I was part of the IP team, excuse me, and we were extremely unhappy.

Gardner: Yeah. Really.

Supnik: But I could sort of tell this was going to happen, because in '96 DEC held its last sales boondoggle. You know, the kind of thing where you get the top 100 salesmen and you take them off and, well, it was a luxurious cruise in the Mediterranean to all of the great archaeological sites around the Mediterranean. In order to make it a tax write-off, you have to disguise it as training.

Gardner: Right.

Supnik: So there would be technical training, and I was the designated technical trainer.

Gardner: Oh.

Supnik: Ah. <laughs>

Gardner: You lucky dog.

Supnik: I was so lucky. It was a wonderful cruise, and it was on a brand-new cruise ship that was only about a third occupied. In typical DEC fashion. You know, the captain invites us for a tour of the facilities and up to the bridge. After a while I said, "You know, this is all really interesting. Can we see the engine room?"

Gardner: Yes.

Supnik: And so yeah. We go down and we see the engineering spaces, and when we come back he said, you know, “I’ve been a cruise captain for 30 years. No one has ever asked me to see the engine room.”

Gardner: <laughs> But DEC engineers are going to do that.

Supnik: They’re going to do it. So one morning on the cruise, I was up on the uppermost deck where there was a running track, and Bob was an avid jogger. I ran into him, and we started chatting and at the end of it, you know, when I asked him, “So what now?” he said, “Bob, I just don’t know what to do. I just don’t know what to do.”

Gardner: That was he said?

Supnik: That’s what he said.

Gardner: Okay.

Supnik: And that-- and then in early ’97, the company was sold to Compaq.

Gardner: Yeah. Well, the board knew what to do.

Supnik: The board knew what to do.

Gardner: Wow.

Supnik: Well, this was of course quite a shock, and it took a long time for the deal to consummate, but we sort of knew that we were going to be acquired, and the Research Group had already had to close its Paris office because of the declining financials. So, we figured that it was more important than ever to get some of our work commercialized, and we thought Compaq might actually be able to do this. They had the reputation of being a marketing powerhouse. They had a consumer division.

Gardner: Right.

Supnik: Well, the first thing they did was take over AltaVista and put some of their consumer people in charge of it, and they promptly remodeled AltaVista as a Yahoo competitor. They made it a media aggregation site, they put ads on it, and within a year Google was born and took the market over. That was the first indication that Compaq had no magic with respect to product innovation. So, the next thing we did was [this]: we had the prototypes ready for the Personal Jukebox, so we packaged up 20 of them and on each of them we had ripped 20 CDs. To make sure it was perfectly legal, we actually included physical copies of the CD, so this was not piracy. We had bought the CDs. We had copied them to what was effectively a home tape recorder, and we shipped it to the Compaq executive committee and the board of directors. This was for--

Gardner: Oh, that's a good idea.

Supnik: We thought so.

Gardner: <laughs>

Supnik: And we heard back from them with, "Thank you very much. These are really interesting. We like them," followed by, "No one's ever going to pay $500 for something that just plays music."

Gardner: Mm-hm.

Supnik: And the design was sold off to a third-rate Korean company which took another year to productize it, by which point, well, it came out before the first iPod but not by much, and it was bigger and clunkier and we had basically wasted two years. They lost two years, during which time Apple--

Gardner: Well, Apple has smaller drive in it.

Supnik: Yes.

Gardner: You could've had a smaller drive in yours.

Supnik: We could've if--

Gardner: Because you weren't doing the drives. They were--

Supnik: Right.

Gardner: --becoming available outside.

Supnik: They became available, but the Korean company didn't have the resources to redesign the product, and by that point, the DEC team had scattered. So, they stuck with the laptop drive. It was big, and it did have some advantages, but they weren't enough. In particular, it had a replaceable battery, so I kept it for years. I was still using it long after I had acquired an iPod because if I went on an international flight before they had USB ports and the like, I could just take a supply of batteries and swap them in and listen to music. Anyway, the prototypes are now in the Computer History Museum.

Gardner: Oh, they are.

Supnik: Yes.

Gardner: Wow.

Supnik: And essentially every other asset in the research portfolio was wasted, and so as Compaq was coming, you know, when we were merged in, I could see what-- I could see the writing on the wall, and so I planned to leave. I had a particular executive arrangement that said, "If you'll stay this amount of time, when you leave, there's a bonus." I stayed that plus one day and then I was gone.

Gardner: <laughs>

Supnik: And I was sorry to see it end that way. Now, of course, it didn't last much longer.

Gardner: Right.

Supnik: Within two years, Compaq sold itself to HP because its business ineptitude had just wrecked both its PC business and everything else. HP became the inheritor of all of DEC's intellectual property. They told Compaq to close down Alpha. They sold all of the IP and the compiler technology off to Intel, and that's…

Gardner: And that's where Alpha disappeared to?

Supnik: That's where Alpha disappeared to.

Gardner: Was there any fab-- after the fab was sold to Intel, did Intel make Alphas?

Supnik: Yes. Intel made Alphas for a while, but because their view of process control was so different from ours, they were unable to yield the speeds. I mean, they refused to run the fab as tightly as we did because the yields weren't good enough, and they wanted to put other products through there. So it took a while for us to move over to Samsung as a foundry, and we probably lost two years in terms of the speed curve, and it [Alpha] never recovered. Now, so eventually HP picked up everything. They picked up VMS, they picked up Alpha UNIX, they picked up all our technology. They did their best to kill it all.

Gardner: <laughs>

Supnik: They did--

Gardner: Unintentionally but--

Supnik: No, no. Deliberate--

Gardner: Really?

Supnik: Deliberately.

Gardner: Okay.

Supnik: They told, like, the VMS customers, "You really want to convert over to HP-UX and the VMS customers said, "Thank you. We'll go to PowerPC. We'll go to Power and AIX if that's what you want." But in fact, many of them stayed and to the-- they even stayed through the Itanium port of VMS and when <laughs> in 2014, yeah, 2013-'14, HP announced end of life for VMS, having disbanded the group and shipped the product off to be maintained in India, which didn't work.

<laughter>

Supnik: There was such an outcry from the customer base, yet again, that they licensed the IP to a group of former VMS developers who were now running a little independent company down in Bolton called VMS Software Inc., and they support VMS releases for Alpha and Itanium. They're porting to the x86, so VMS lives, which is truly amazing, given all the attempts to kill it that occurred. So for the first time in 22 years, I was out of a job and again, I took the summer off. This is 1999. But I was only 52 at the time. Fifty-two. Yeah, 40 from 90 is 50. Right, 52, and I wasn't used to being idle. My wife wasn't used to me being around the house.

Gardner: <laughs>

Supnik: So I had some incentives.

Gardner: I think the term is, "I married you for better or worse, but not for lunch."

Supnik: Yes.

Gardner: <laughs>

Supnik: So I started looking around and I'd pretty much decided that after 22 years in large companies it was time to do something completely different. To go to a small company, startup if possible, and get away from bureaucracies.

Gardner: Yes.

Supnik: So, I didn't look very hard when I got invited to be the VP of Engineering in a small startup called FairMarket. These were the early days of the internet auction market, before eBay had consolidated everything, and lots of people were doing auctions. FairMarket was basically a service bureau for people who wanted their own auction sites, and furthermore they produced a consolidated view so that all the auctions were melded into one large auction to get more volume. There were maybe 20 people. They were in this tiny basement in Woburn, but they had just acquired a substantial amount of funding from a VC, and they were ready to start growing, so they needed professional engineering. So thought, "Sure, why not? Internet startup. What could go wrong?"

Gardner: <laughs> Yes. This is that time--

Supnik: Yes. This is the bubble.

Gardner: --when-- the Internet bubble. Yes.

Supnik: Right.

Gardner: Okay.

Supnik: This is the Internet bubble.

Gardner: Yeah. That just dawned on me. <laughs>

Supnik: So I became the VP of Engineering, and it was interesting, because unlike every other Internet startup on the planet, they weren't trying to use Sun technology and Unix. They were using Windows technology for its cost advantages. But now we face the problem of how to scale, how to grow, how to make this structure of scripts into something that was robust ,and we were trying to do this just as the first wave of Internet Windows viruses started to hit. Well, that was among the most exciting two years of my life that I can imagine because I was not only VP of Engineering, I was VP of Operations, Computer Operations. For the first time ever I slept with a cell phone by my bed, because we would have outages at 3am in the morning when something snuck onto the system. It was terrible. It's not that NT was unreliable. NT was a fine operating system. It's not that SQL server was a bad database, it was a wonderful database. But oh, my God, the security was brutally bad, and we were trying to move to a bigger data center. We were trying to scale up all these things. All these things were happening simultaneously. It got to the point that it was so bad that I, through our Microsoft rep, said, "I want to see the guy who's running Microsoft security in my office," and they sent him, and he was an ex-DEC person.

<laughter>

Gardner: Okay.

Supnik: He had been in charge of a project at DEC to do a secure version, an A1 secure version, of VMS, so the highest DoD classification, which involved making a virtual machine monitor on the VAX that would run VMS in the virtual machine. Very interesting project, so he knew his security, and he was apologetic as hell for what was going on, <laughs> and all he could say was, you know, "It will get better but it's going to take time," and it did. So the place [FairMarket] grew and grew very rapidly. We were 20 people when I got there and it seemed like [only] a matter of months and we were 150. We--

Gardner: Who were your investors?

Supnik: There was just one investor from the west coast, whose name I don't recall. I'm sorry.

Gardner: Okay.

Supnik: But everything I had hoped about a small company proved true. Both the good and the bad. So for example, on my first day, and I'm just settling into my desk, they come and say, "We've lost Internet connectivity. The firewall is down. Can you fix it?" I've never done that. I've never dealt with a firewall in my life, and it turns out it is a variant product from DEC that got labeled the AltaVista Firewall. So I was able to restart it and get it running again, and that made me realize that in a small company there is no support infrastructure. You know, you want to do something, you do it. The good side of this was when we had laid out a plan for expanding the data center and the computer infrastructure, and it amounted to like three-quarters of a million dollars in capital. Huge purchase for a company of that size. So I write it up and I take it up to the CFO, and he asks me to describe it. I give him a two-minute description, and he looks at the PO, and he signs it, gives it back to me, and I say, "Okay. Thank you. What's the approval process now?" He said, "You've just been through it."

Gardner: <laughs>

Supnik: I said--

Gardner: You said, "This is good."

Supnik: I said, "This is good."

Gardner: <laughs> Yes.

Supnik: "This is very good." And so the company, you know, rode the Internet wave, went public. The stock was offered at 15 and popped immediately to 45.

Gardner: Oh, my goodness. Yeah.

Supnik: Course, we were all locked up, and we couldn't take any advantage of it.

Gardner: <Inaudible>.

Supnik: And during the lockup period the bubble burst, and I mean--

Gardner: Of course.

Supnik: --"Boom." And by the time our stock options were usable, they were below water, and the company started laying off people because the revenue wasn't there, and we were now public. As we started shrinking back from 150 to 80 heading toward 60, I went to the CEO and said, “You don’t need me anymore. I’m overkill for what you’re going to be. I’m going to leave.”

Gardner: Mm-hm.

Supnik: And it was pretty amicable. I mean, they understood, and I had trained a successor, a fellow from DEC, from the DEC UNIX group, named Jeff Myers, who took over for me. He later went to Chronus, where he was VP there. So it’s 2001, post-bubble, looking for something else to do, and so I went around to all the VCs and started talking to them. The local VCs now, not the west coast ones, and the people at Charles River said, “Oh, we have a new startup. You should go talk to them.”

Gardner: Who were you talking to at Charles River?

Supnik: Oh, cripes. I’m about to lose his name. He’s the guy who did storage and networking. He’s not there anymore. [Chris Baldwin.]

Gardner: Okay.

Supnik: The three investors were Charles River, Matrix Partners Andy Markowitz, and then there was a third smaller one.

Gardner: Okay.

Supnik: My head for details is evaporating.

Gardner: That’s okay.

Supnik: So I met the team, the startup team, which they had been one of like half a dozen fast Ethernet switch startups, and in the mid-‘90s and theirs, like, every other one, was successful. I mean, it’s amazing. All six of these fast Ethernet companies ended up, switch companies, ended up with successful outcomes. These guys ended up being sold to Lucent, Lucent Alcatel, or maybe it was just Lucent at the time, and stayed there for a while and we said, “Yeah, it’s time to try something different.”

Gardner: What was the name of this company?

Supnik: The previous one?

Gardner: No. Yeah. Yeah, the one that you joined.

Supnik: Oh. Well, at the time it was called Breakwater.

Gardner: Okay.

Supnik: It was going to be Breakwater Networks, because the CEO Josh Weiss was an avid sailor. He had a beautiful 44-foot sailboat that he kept on Narragansett Bay, and it took a while of interacting with the VCs before one of them said politely, you know, "Breakwater, I understand the nautical connotation, but you just don't want that for a company name. There's just too many ways you could make bad jokes about it." Okay. So it was changed to Nauticus Networks.

Gardner: Okay. <laughs>

Supnik: And I joined in 2001, again as VP of Engineering. Tiny little startup crew. Steve Metzger was the Director of hardware, Paul Philips was the Director of Software, Brian Ramelson was kind of chief architect. All of them from the same startup, all of them network veterans. I had never worked in a networking company before. I never worked in networking at DEC. So--

Gardner: Good. Interesting.

Supnik: Interesting. You know, never worked in the Internet before, doing internet startup. Never worked in networking, doing network startup.

Gardner: Well, it's fun. You're leaning new stuff.

Supnik: I am.

Gardner: <laughs>

Supnik: So it was well funded and the idea was to do a-- given that the Internet was ballooning out of control, was to do a load balancer at levels of performance that had never seen before. Most load balancers were PCs with some software on it. This was going to have a hardware engine for doing TCP. Now, something [like] that is fairly commonplace today [in the form of TCP offload engines], but at the time, [it\ was just completely revolutionary. Furthermore, it was going to be a proxy, so it would send connections and regenerate them, which would permit you to have full SSL outbound [everywhere].

Again, well before anybody thought of being able to put SSL everywhere. Now, you might argue that we were being a man in the middle of an SSL stream, which is not considered good, but it was on premises and therefore theoretically physically secure. Now, that's what we were doing. So we went and recruited a team of people we knew. ASIC designers, CAD people, software designers. A woman named Karen Wise [as] director of software. Wonderful person. The whole team was extremely good. I really, really liked all of them, and it was a big, complex project. To get the speed, we needed network processors on the front end, which was a great concept but impossible to program.

<laughter>

Supnik: And we had this TCIP/IP engine, which would be supervised by a controlling microprocessor. In theory, even the load balancing policy decisions would be done in hardware, so that we could do the unheard of idea of more than 50,000 connections per second through a single switch. Through a single load balancer, and they would be, you know, available in pairs. They would do all the proper things, and it turned out to be a lot harder than it looked. In particular the ASIC design to do TCP/IP in hardware. We actually were using field programmable gate arrays. It was very complicated, and we had some brilliant CAD technology. There was an engineer I had known from DEC named Wilson Snyder, who had… (again, today it seems commonplace) but he had a system for driving regression testing automatically that would build hardware, build software, put them together, run them through a distributed simulation farm overnight, collect the results ,and automatically correlate failures back to the changes made the previous day. So, you'd come in in the morning, there'd be mail waiting for you saying, "You broke this. Fix it." It was pretty rude but, you know, we were all friends, and again, the same thing happened.

The place grew from the original six or eight of us out to 40 and then 80. Built a QA department, got some marketing and product management, and finished the product and then realized we had made a serious mistake in our assumptions. Because it turned out that at the time, this is the early 2000s, the market for a load balancers of that capability was like four companies. Google, Hotmail, Monster and maybe one other. [Yahoo] There just weren't enough sites getting enough internet traffic to need the kind of capacity we had built ,and we had built many other interesting things. It was virtualizable. You could partition it, into four separate zones that ran independently of each other. Nobody thought that was interesting. So we got up to the C round, and the VCs had already begun shopping the company. Sun was interested. Sun still did networking at that time. They liked the virtualization and security aspects, and they were proposing, they were dickering around a price that would've left the VCs whole. It wouldn't have made much money, but…

Gardner: Mm-hm.

Supnik: So the VCs decided not to do the C round, and they told Sun that.

Gardner: They told what?

Supnik: They told Sun that they weren't going to fund the company. At which point Sun's offer dropped to five million dollars from fifty. Well…

Gardner: Yep. And so now they aren't going to get their money back.

Supnik: They aren't going to get their money back. For about three months, we struggled in limbo on a bridge loan from Sun, because Sun eventually did up its offer to something better, but the VCs were never made whole. All the executives were on no pay or quarter pay so that we could pay the staff out of this bridge loan, but eventually the deal closed, and Sun did buy us. It was closed at the end of, I want to say, 2004.

Gardner: Mm-hm.

Supnik: Or maybe it's 2003. Yeah, 2003. It's 2003, sorry. And as of January 1st 2004, we were all of a sudden [Sun] employees. It's going to take a while for them [Sun] to make room for us over in the [Burlington] campus, so we continued to operate almost as an independent company for the first four months, and then in May of 2004, we pack up our, you know, our belongings. We move over to the Burlington facility. Only to find that the vice president who acquired us has been fired, the division he ran has been disbanded, Sun is intending to get out of the networking business, and we are now part of a new effort to put Sun in the PC and PC server and Blade server business. Run by none other than Andy Bechtolsheim.

Gardner: Oh.

Supnik: So Andy had done a startup for beautifully packaged servers and Blade servers with very fast backplanes. He was into InfiniBandband long before anybody else, and Sun needed hardware resources and software resources to make his vision a reality, so we walked in there and suddenly we were PC designers. Well, I actually didn't feel that there was anything that I could usefully contribute. They didn't need a VP of Engineering because Josh Weiss, the CEO, was now the engineering manager of this group, so he didn't need me. I ended up getting hooked up with Greg Papadopoulos, the CTO, and Greg basically shanghaied me off into the Sun Research group.

Gardner: Oh.

Supnik: Where I was, you know, supposedly going to look around for something to do and so forth, and then I got to make some trips to the west coast. I met the [Sun] people, and by the fall of 2004, I had concluded that even though Sun had the best gym in the world and I was using it a lot, and I was in better physical shape than I was ever in my life, this actually wasn't a career.

Gardner: Right.

Supnik: And--

Gardner: And they didn't really have anything for you to do.

Supnik: They didn't have anything. Their research group was a good group, smart people, but they weren't interested in what I was interested in.

Gardner: What were you interested in?

<crew talk>

Supnik: Well, what I was interested in was new technology. So one thing is that at that time Sun was very tight with AMD, and AMD was looking to build more capable chips. They had basically stolen a march on Intel by going to a 64-bit architecture while Intel was still pushing Itanium.

Gardner: Right.

Supnik: [AMD] They had gotten rid of the front side bus and moved the memory to the back of the chip for much better performance, an idea they stole from DEC. Excuse me. Excuse me. An idea which DEC did first and which they--

Gardner: Borrowed.

Supnik: Yeah, they borrowed. It wasn't patented, and it was--

Gardner: Yeah. There's nothing wrong with that.

Supnik: And they hired the key designers who did it.

<laughter>

Gardner: That's even smarter.

Supnik: Right. Yeah. A fellow named Jim Keller [and Pete Bannon]. And they were talking about adding more of these interprocessor links. They had two or maybe three at the time, so it was easy to build a four-way, and with some pushing you could build an eight-way, and they were talking about adding a fourth link. So I started looking at a project to take the Blade server technology, and instead of using it to make clusters, to use it to make symmetric multiprocessors on an enormous scale, and in particular I was looking at the concept of having 32 blades-- no, sorry. Not that many. It would've been 32-- 8 blades, 4 processors each, 4 cores each, so it was 4-by-4-by-8.

Gardner: Okay. Four cores in each. Yeah, in each <inaudible>.

Supnik: Yeah, so 4-by-4; 16-by-8 is 128. So would've been a 128-way SMP, and I thought that fit Sun's business model pretty well, because they had technology for thread scaling beyond most people, and so that's what I was interested in, but Sun wasn't. They just didn't want to build a big machine like that. They were very invested in their-- the technology they had bought from Floating Point Systems, that had been included in the Sun 10000, 15000 and 25000.

Gardner: Yes.

Supnik: So in the late summer, I got called by some old DEC colleagues, and they were trying to build a large-scale, high-performance technical computing machine that would be very low-power. They were going to use multicore technology and a very interesting new interconnect, and they had been moving along on some very, very threadbare angel funding. It was Jud Leonard, who had been one of the principal PDP-10 architects. Matt Reilly, who had been one of the innovative designers at DEC Semiconductor, and an old clever business guy named John Mucci, who had left DEC in the early '80s and banged around the high-performance technical computing industry. Smart, funny, and very, very cunning.

<laughter>

Supnik: And it was a good idea. But it took them too long to get money. They had started trying to get money in 2002. It was not until the summer of 2004 that they began to see a glimmer of real funding, and by then Intel had caught on, and AMD had caught on, to the idea of multiple cores. So it was a little, they were, we were, they were just a little late, as it turned out. But it was a fascinating project. Jud had designed this unique interconnect based on the most obscure graph you had ever seen. The result was that the machine scaled in powers of three. Not powers of two. Powers of three.

Gardner: Huh. Okay.

Supnik: So you ended up with these bizarre processor counts, but you could show that it was damn near optimal for a given investment in wires and silicon imports. So we got money from Polaris. Bob Metcalfe was on the board. Oh, jeez. Again, I won't be able to recall the others. There were two others.

Gardner: Mm-hm.

Supnik: And there was a lot of skepticism as one of Bob Metcalfe's partners said to him, "You're really not a VC until you've lost a fortune on a systems company."

<laughter>

Supnik: And indeed, there had not been a systems company startup in years. But with the government investing again in high-performance computing after a long lull, the post-911 increase in defense spending had led to a huge uptake in high-performance technical computing spending. And it was also driven not just by the government and the intelligence agencies but by advances in computing for chemistry in pharmaceuticals and so forth.

Gardner: Mm-hm.

Supnik: It looked like there was a real market. If we moved quickly. So we got money, started building a team. One of their next hires after me was a fellow named Larry Stewart, who was sort of a jack-of-all-trades, master-of-everything researcher out of the Cambridge Research Lab, who was one of the technical founders of Open Market.

Gardner: Okay.

Supnik: He had made a fortune there and was now looking for something to do. One of the brightest guys I'd ever met. You hand him any problem, in any discipline related to computers, hardware, software. It was just, he just goes and does it. He had been one of the principle contributors to the so-called Alpha development system [Alpha Development Unit, or ADU], which is a brilliant design that he and Chuck Thacker did with Dave Conroy. So we went off and built a small chip team and small software team. So let me recapitulate. John Mucci as the business head, and Matt Reilly and Jud Leonard as the technical brains, had been trying to get this little startup off the ground for two years when they finally got funded in 2004. I came on board as the VP of Engineering. Larry Stewart joined almost immediately as from-- as architect and sort of software honcho, and we built a relatively small team. I think the company maxed out at 40 <laughs> in terms of, at least when I was there. And about six months into this, you know, when we're trying to build a system even bigger than this multi-processor I had envisioned at Sun, where it's going to have 36 boards on a centerplane with 5,872 processors. It's like, as I said, it goes in powers of three, so it's a--

Gardner: Yes, yes, I understand.

Supnik: It's a bit weird. I went to John and I said, "John, if Sun were doing this, they'd have 400 people on this project. We're going to have less than 40," and he just smiled and said, "Sun 400, SiCortex 40, advantage SiCortex." How right he was. So again, it was being, you know, the jack-of-all-trades. We were doing chip design. We were doing system design. We were doing software design. We were porting Linux. We were getting compilers. We were creating application structures and run-time libraries. We're doing firmware, doing board design. It was truly a stretch experience for all of us, and it's not surprising that it ran a little late, but for one thing, the team was almost all ex-DEC people. None of them had ever built a semi-custom chip before. They'd always been full custom. So doing semi-custom, i.e. writing in Verilog and generating logic and then automatically laying it out, was a new experience. And in particular, [there were] things that the tools of DEC could do that they were used to and that the commercial CAD suites could not do. So it took a little longer. Mine was, probably the most important contribution I made, was to convince John that we could not go to market only with a big machine. That the number of customers who were going to shell out a million and a half to two million dollars to a startup was probably limited to one and it had three letters and it's--

Gardner: Yes, it had three letters and it lives in a Fort in Maryland.

Supnik: Yeah, exactly.

Gardner: Yeah, we know that one. <laughs>

Supnik: So I convinced him to do a much smaller machine, a machine that could be bought for $150,000. Eventually, we stretched the same design from a workstation scale machine that had 72 processors through one that [had, that] was one-ninth scale, one that was one-quarter scale, and then the big machine. Then Larry Stewart, bless his heart, figured out how to insert continuity cards, because the network that Jud had designed, the interconnect network, required everything to be there. He figured out how you could remove cards and substitute much cheaper continuity cards and still keep the network whole. So we could now sell the system by the slice. You could get the big chassis but you could buy, like, instead of like 36 boards, you could buy nine to start with and expand it as you need to. So there's a lot of brilliant technical work. Eventually we bought a superb complier team that had been for some reason at QLogic.

Gardner: <laughs>

Supnik: Was a very friendly transaction. The QLogic president, who I think has since passed on, said, “Look. I love these guys.” They had been part of a startup that was trying to do an InfiniBand-based HPTC offering. Said, “I love these guys. I’m not going to let them, you know, just twist in the wind. Let’s make a fair deal that gets them, that keeps them whole, helps us a bit, but above all gives you a compiler team dedicated to your needs. They were very, very good. But because the startup had begun late and then had taken a little longer to get the first machine running, by the time it hit the market, the HPTC market had consolidated around cheap Intel multiprocessors tied together with InfiniBand. That’s what you bought. You wanted to build an HPTC center you went to an OEM in Taiwan, you said, “Give me 64, 128 or a thousand 1U servers with two Intel chips in them and some memory.” You went to Mellanox and you said, “Give me an InfiniBand switch big enough for all these guys.” You tied it together, you ran MPI on it, and suddenly you were in the super-computer business. Now, we still had an enormous power advantage. And we still had a connectivity network that could run rings around anything.

Hendrie: Right.

Supnik: But now we were in a niche and not the general market.

Hendrie: Mm-hm.

Supnik: And so it didn't work. I left in- in early two-- no, mid-2008. And the company hung on for another two years, I think, before it just folded up and its technology got sold off. The only two really large machines were sold, one to a university under very favorable terms, and one to a three letter agency.

Hendrie: Mm-hm.

Supnik: And the one that was at the university is now at the Computer History Museum. So--

Hendrie: Really? Very good.

Supnik: And the little workstations that we built, which were an afterthought, are in intense demand because if you want to have, you know, something on your desktop with which you can experiment on a networked supercomputer-like machine with lots of processors, it's still about the best you can do. It's not fast, but if you want 72 processors in the size of a mini tower.

Hendrie: Mm-hm. We can get it.

Supnik: So they don't-- they don't turn up on eBay often and when they do, they go for really good prices.

Hendrie: <laughs> Wow.

Supnik: So it's now-- it's now--

Hendrie: Oh, I have to-- full disclosure. I now remember I looked at that company as a VC.

Supnik: Mm-hm.

Hendrie: And--

Supnik: That's okay.

Hendrie: Yeah. They were just talking about you at that point.

Supnik: Right.

Hendrie: I don't think they-- this was probably in those two years when--

Supnik: Yeah.

Hendrie: -- they hadn't gotten any funding.

Supnik: Right.

Hendrie: And they were-- they were talking about you, but you weren't there.

Supnik: I wasn't there.

Hendrie: You weren't there anyway. Right.

Supnik: And I don't regret it. I mean that was-- it was just an immense amount of fun.

Hendrie: It was an interesting project. It really was.

Supnik: It was. And I'm sorry we couldn't make a return to the investors, but the idea was right. These days, you could take Intel's mobile chips, because they're low enough power now - and an onboard interconnect probably built on InfiniBand for now, but Intel is going to have it on chips soon enough. And you could build a SiCortex machine out of industry standard parts to run industry standard software.

Hendrie: <laughs> Look what happens.

Supnik: Wouldn't be a problem.

Hendrie: Yeah.

Supnik: So it's now late 2008, I'm out of a job. I'm only 60 years old.

Hendrie: You're still not ready to retire.

Supnik: Still not ready to retire. And it's the Great Recession. The bottom has fallen out of the universe.

Hendrie: Oh yes. That's true.

Supnik: So I started networking with people, and I ran across an old colleague from the Semiconductor Group by the name of Rich Marcello. Rich had started as a chip designer and then a supervisor on Rigel. But he had the foresight to switch into management and into software. And then [he] ended up running the VMS Group for DEC and then for HP, moved on to other things. And now he was the business VP-- the business president, sorry, of one of Unisys's three divisions. Unisys had three divisions. It had a systems integration business and outsourcing business. It had a group that did vertical applications, and [it] had a group that serviced the old mainframes-- that designed and serviced the old mainframes. One of those made money, the others did not. You can guess which one.

Hendrie: Yes.

Supnik: So Unisys was in desperate straits. They were-- they had a lot of debt. The credit markets had dried up. And they were looking to spin out the mainframe business, sell it as an independent company to raise cash. And Rich wanted to know if I was interested in being part of that, 'cause he felt he needed different engineering leadership. I, you know, I said, "Nah, not particularly, but keep in touch." Well, in September of 2008, the Unisys board came to its senses, fired the existing CEO, and for the first time in history hired somebody from the outside, not another Unisys insider.

Hendrie: Mm-hm.

Supnik: And the new president, new CEO looked at this and said, "You know, this is the only part of the company that makes any money."

Hendrie: Yeah.

Supnik: "I'd be absolutely crazy to sell it off."

Hendrie: Right.

Supnik: And he kiboched the deal.

Hendrie: Mm-hm.

Supnik: Quite rightly. So, you know, it's now December-- getting onto December. I have not been able to find any work. And the winds of the recession are blowing on.

Hendrie: Yes.

Supnik: I'm- I'm nervous.

Hendrie: Mm-hm.

Supnik: And I don't want to retire. So I go back and say, "Well, you know, you said you needed new engineering leadership. Is that true?" Rich says, "Yeah, I do. So there's no real prospect. The company is in dire straits. You know, you could be out of a job in three months. But, yeah, if you want-- if you want it, sure."

Hendrie: Okay.

Supnik: So I went down to the headquarters in Pennsylvania and interviewed with some of the people there. And I could see the company was in dire straits. They had this headquarters building, lavish. You know--

Hendrie: This is in Blue Bell.

Supnik: Pennsylvania.

Hendrie: Yeah.

Supnik: In Blue Bell.

Hendrie: Yes.

Supnik: And, you know, executive suites, heavy rosewood furniture. I get interviewed in the executive private dining room, which got closed the next week, by the new president who just had no time for such stuff. And he then proceeded to sell off that building, 'cause he thought it was completely inappropriate.

Hendrie: Didn't need the building either. Yeah.

Supnik: Right. And- and we agreed we'd-- I'd be hired. This division, the systems division was headquartered out of Nashua, so I didn't have to go to Penns--

Hendrie: Oh, all right.

Supnik: I did not go to Pennsylvania all the time. And it turned out engineering was in Pennsylvania, Minnesota, California, China, India, and Australia. i.e.. every place except where I was. So I knew there would be a lot of travel. Okay, that was part of the term-- part of the game. It was a job and a good job. And in particular what they were trying to do I thought was interesting problem. Unisys had built CMOS mainframes for years. But as their market had shrunk and their resources had shrunk, they had concluded that they could not afford to do hardware design. And they were trying to get their software to run on the x86.

Hendrie: Mm-hm.

Supnik: Rich knew this part of Alpha and overseeing the effort to move the VMS software base onto a new architecture, so all right.

Hendrie: Yeah, he said. Yeah.

Supnik: Yeah. Yeah. So I thought it was a very resistent problem. And so I took the job, went out to meet people. And I had never run into a more demoralized, frustrated, and almost mutinous group of people than the engineering cadre at Unisys.

Hendrie: Really? Yeah.

Supnik: Over the previous decade, the place had been literally decimated. It had been 3,000 engineers at the beginning of the decade. When I got there, there was 600 left in the U.S. They felt that the eng-- that they had kept the company alive with- with their mainframes, and that the company did not give a damn. They had never heard an executive say a good word about them.

Hendrie: Wow.

Supnik: What they were loyal to were the customers. And they were loyal to the customers in a way I had never seen. When a customer calls up and says I have a problem, it doesn't matter what the engineer is doing, if it's his area, he drops it [his work], he works on the customer problem till the customer's whole. If he has to stay up 24 hours a day, he'll do that. It was just amazing. And I realized--

Hendrie: The customer focus. Yeah.

Supnik: The customer focus. We had something to build with. We have something we can build with. Now the question was whether the company was going to survive long enough to try and do any building.

Hendrie: Yeah.

Supnik: But the new CEO and his VP of finance wrought a miracle. They refinanced the debt. They had to pay exorbitant interest rates, but they refinanced the debt, and they got the debt payments moved out of 2009.

Hendrie: Mm-hm.

Supnik: They did a ten-for-one reverse stock splits so that the stock, which had bottomed out at 27 cents, was listable on the New York Stock Exchange. And the CEO just said, "We're cutting $250 million of expenses. It's going to be brutally painful, but guys, here's how much money comes in, here's how much money goes out if we're going to be profitable again"--

Hendrie: <laughs> Had to be the same.

Supnik: Right. And he was incr-- and what-- and as-- what I liked about Ed [Coleman] was that instead of having these elaborate strategies, with him it was just simple stuff. It was focus on the customer, deliver products they want to buy, cut expenses, you know, be operationally good. It- it was-- it was something you could understand. It wasn't some elaborate transformational strategy. It was--

Hendrie: Yeah. Yeah. Right. It didn't require magic.

Supnik: It didn't require magic.

Hendrie: Those are the ones that work. <laughs>

Supnik: Yes. And he coupled that with a personality, the most egoless CEO I have ever seen. He just had no time for the trappings of being a CEO. He grounded the- the air fleet. There were still two helicopters and a private jet. He grounded them and sold them off or let their leases lapse. He did away with, you know, expensive hotels. His idea of a good meal out with- with one of the executives or with me was we'd go to the local chain restaurant in the Paoli shopping center, you know, and spend 40 bucks between the two of us. And he didn't drink. He was-- he was-- I think he was probably the finest CEO I've ever worked with. I would've-- I-- as a result I would've followed him, you know, anywhere.

Hendrie: Yeah.

Supnik: So for my part what I found was that rather than being a test of my technical skills, Unisys was a test of my management skills.

Hendrie: Wow.

Supnik: And that the lesson Peter Conklin had taught me on Alpha about enrollment management was, in fact, the key. That I needed to get the teams enrolled in the goals of being a good engineering group again. Of executing the strategy of getting off proprietary hardware onto the x86 and on making our products more attractive. Of branching out into new areas and understanding that the world had changed and, you know, —[that] resources were going to scarce, we're in the middle of the worst recession in our memories, and we were going to have to not only survive but thrive. So within a year I had worked through with the teams that we would no longer build hardware. It's not that we knew exactly how we were going to get the highest performance levels we needed out of running on an x86, but we would make the leap of faith and stop building hardware. And just like a DEC, this meant that a group got laid off.

Hendrie: Yeah.

Supnik: And just like a DEC, because they had been included in all the decision-making and all the data gathering, and [everything] had all been done in the open, they were okay. They weren't thrilled, but they-they didn't leave [with], you know, cursing the company under its breath.

Hendrie: Right.

Supnik: And then the second thing [that-] that I did, was I restarted college recruiting, which the company had not done for seven or eight years. Because, of course, it was laying off engineers left and right. But everybody in Unisys looked like me. The average age of the engineering group in Pennsylvania was 52. The average age in Minnesota was 55. And it was easy enough to draw a graph that said, this is when the lights have to be shut off because nobody is there any more.

Hendrie: <laughs> Yeah.

Supnik: And so even though we had very little money and we had no [cre--] credence or cachet [on-] on college campuses, by 2010 we were recruiting at colleges again. Started with one or two-- actually it was two-- late 2009 when [they-] they came to us. And we were able to get some because, of course, times were desperate back then, and college students would take [any-] any job that seemed like it would pay.

Hendrie: Right.

Supnik: And we hired local people. So in Minnesota we hired from the University of Minnesota. From-- for Pennsylvania we hired from Drexel.

Hendrie: Mm-hm. Yes.

Supnik: We restarted the-- our Co-op Program with Drexel. In California we went to, not to the UCs, but the state colleges instead.

Hendrie: Mm-hm.

Supnik: So we, you know, we got a handful the first year.

Hendrie: Yup.

Supnik: And then they were able to go back to campus the next year and say, "You know, this is a really interesting company. What's interesting is when I get hired I'm not, you know, some junior trainee. As soon as I learn my way around, I'm doing something real, something significant." Because we didn't have any way of buffering them [from-] from real work.

Hendrie: Yeah.

Supnik: So that was the second major initiative. The third thing was to try a very different field of endeavor. So we decided to build a cloud and cloud products. We had no experience. We didn't know anything about it. The company--

Hendrie: It was clearly something that was growing--

Supnik: It was growing.

Hendrie: -- and more and more people were using it and blah, blah, blah, blah. Yeah.

Supnik: Right. And the [com--] the company [actually, the Systems Division] had been working on something called Real-Time Infrastructure, which was clearly going nowhere. But it gave you certain basic capabilities that looked like they might be useful in the cloud. So [I] formed a team, staffed it from whoever was willing, and gave it an impossible deadline.

Hendrie: Mm-hm. <laughs>

Supnik: Six months later, they had a functioning cloud together. Now there was actually a lot of pushback about this because in a mainframe company, you know, there's no such thing as good enough engineering. It has to be perfect. Customers of a mainframe are spending tens of millions of dollars for something that never goes down and never crashes.

Hendrie: Right.

Supnik: The Internet has different expectations. And so [I-] I started blogging. And part of the blogging was to communicate to the group that there were different ways of thinking about these things. And <snaps fingers> I remember the blog about when the cloud was ready it was titled, <clears throat>" It May Not Be Victory, But It's Good Enough." And- and so that was-- that was another major thing. And then we decided to draw on the company's expertise in security to try and do some security products.

Hendrie: Mm-hm.

Supnik: So we were trying to expand the portfolio while trying to revitalize the group, and people just rose to the challenge. I was-- I was really, really pleased. You know, by 2014, we had the team that was descended from the old Burroughs line, like as if problems weren't complicated enough, Unisys actually had two lines of computers.

Hendrie: Yes, 'cause they acquired Burroughs. Yes, at some point.

Supnik: And- and UNIVAC. And so they had a-- I- I used to say that I'm so proud, I'm the only person in the world to own not one, but two product lines with non-power of two computers, 48 bits and 36 bits.

Hendrie: <laughs> Yes.

Supnik: And--

Hendrie: <laughs>

Supnik: So by 2014, the 48 bit group, the Burroughs group, had matched the performance of the fastest hardware they had ever built, on an x86 using translation. And they could see their way forward now to growing on the software curve instead of a hardware curve, which is-- means that they had a hell of a lot of headroom for future- future performance growth.

Hendrie: Yeah. So how had they done that? What was the fundamental approach they used?

Supnik: Well, so binary--

Hendrie: I mean they built an emulator? I mean talk to me about it.

Supnik: They started by building an emulator way back in the '90s, and they built it on Alpha, because if you're going to emulate a 48 bit machine with- with tags--

Hendrie: It's nice to have a few extra bits--

Supnik: It's nice to have a few extra bits.

Hendrie: -- rather than making them up. <laughs>

Supnik: But they recognized that emulation can only go so far, so they started adding dynamic translation. And just like-- just like the Java run-time systems started as purely a byte code emulator, and then started growing a just-in-time compiler, so they did the same thing. They said the instructions for [the—for] the Burroughs machine are basically weird byte codes and we will generate native x86 code. And they started by generating very simple, you know, well, I know what this emulated instruction will do, so instead of having a full decode loop I'll just call the run-time. And then when the instructions were simple enough they started compiling simple sequences in-line and using threaded code.

Hendrie: Oh.

Supnik: And then they started applying compiler techniques to their threaded code to make true in-line sequences. Then they started compositing Burroughs instructions into [mult--] multiple functions, 'cause it's stack machine.

Hendrie: Yeah.

Supnik: So there's a lot of very stereotyped code sequences, like push, push, add, pop. No point actually emulating the pushes.

Hendrie: Right.

Supnik: You might as well just say A equals B plus C.

Hendrie: Yes.

Supnik: Lot faster. And so basically [they]- they followed the train of compiler technology, not using any particular backend, but their [own-] own invention. But [when I'm--] when I retired they were now into trace scheduling. So they were sort of getting really up there in terms of the kind of technology they were applying. And so their soft-- their performance on a fairly constant frequency [hardware], because the Intel machines weren't getting much faster--

Hendrie: Right.

Supnik: -- continued to grow. And because the Intel machines had so many cores available, they could have more cores than they used to have.

Hendrie: So they can start, yes.

Supnik: Scaling out.

Hendrie: Scaling out. Right.

Supnik: And the Intel systems with the backside memory bus have much faster memory systems than the old mainframes, so they have the advantage of that memory is much closer.

Hendrie: Mm-hm.

Supnik: And so the performance just grew by leaps and bounds. I think we were at, you know, they used the indus-- the IBM MIP as their basis. And I think the first machine that they released when I was there was 200 MIPS, and then when I left they had just hit 750 on a [single-] single emulated core and had gone from 8-way to 16-way. So they had just built a machine that was order three times faster than anything they had ever shipped. And they did that in the space of six years.

Hendrie: Okay.

Supnik: That was real--

Hendrie: With-- yes, with commercially available processors.

Supnik: Systems.

Hendrie: And now did they use Intel's systems or they're--

Supnik: Intel systems.

Hendrie: Okay. So they even ported to Intel systems, not just Intel processors--

Supnik: Right.

Hendrie: -- and build their own hardware.

Supnik: And--

Hendrie: So, no hardware.

Supnik: No hardware.

Hendrie: Okay.

Supnik: And they implemented something that I had [had in--] had in my mind [since-] since DEC, which is a functionally decomposed system. I had always felt that you could gain some advantages if you decomposed a standard system into computation. I/O. and networking as separate processors, when processing elements. And because all these mainframes were channel-based machines. it was an easy evolution to say, okay, we'll have a processing head, which is an Intel system running emulated code, but we'll now have these channel processors grow [into-] into being full-scale I/O processors and/or network processors--

Hendrie: Right.

Supnik: -- to run the disk and the network.

Hendrie: Yeah.

Supnik: And we'll make sure, going back to the lessons that I learned in the '70s at DEC, that these machines are not in the data path. That the data path, you know, everything flows automatically.

Hendrie: Right.

Supnik: And they did. So the typical machine that they build [is-] is basically a LEGO piece. It's got, you know, one or two processing heads. It's got two to four of these I/O things. You can cross connect them in all kinds of redundant ways, you can grow horizontally as well as by scaling the number of cores. So it's a very, very flexible architecture. It's lovely. And, you know, [they were--] they worked out a lot of that for themselves, so full props to the teams.

Hendrie: Okay.

Supnik: The- the 36 bit machine was trickier.

Hendrie: Oh, could I just ask one more question about the 48-- the Burroughs machine?

Supnik: Sure.

Hendrie: So I'm a customer, and I'm running, you know, Burroughs' code, and the reason I want a faster processor is my business is growing, and I do not want to rewrite all my software.

Supnik: That's--

Hendrie: Fundamental--

Supnik: Funda--

Hendrie: -- the fundamental that's imperative here from the customer's point of view.

Supnik: Right.

Hendrie: What do you give me when you upgrade me to a, you know, to a-- to a faster machine?

Supnik: I gi--

Hendrie: What do I physically get and what do I-- what software do I get? And stay over here.

Supnik: You get a collection of Intel machines tied together to look like one computer. They're running the Burroughs operating system, which was called MCP or Master Control Program.

Hendrie: Yeah.

Supnik: And you could basically disconnect your storage from your old system, hook it up to your new system, hit boot, and everything just runs. Runs faster.

Hendrie: Okay. Okay.

Supnik: Now Burroughs always insisted that customers recompile periodically, because they reserved the right to keep the instruction set moving. And so basically, every five years by contract you're required to recompile. While [we were there-- while] I was there, rather, they actually did one of those transitions where they made a new version of the instruction set that was much more tailored to the idea that it was going to be translated. And if you would recompile to this new version of instructions, of Burroughs instructions, you got[-- you had] another performance increase. But because the old instruction set was still there, if you didn't want to--

Hendrie: You didn't have to.

Supnik: You didn't have to, and you still got a performance increase.

Hendrie: Okay.

Supnik: So it was really a lovely strategy from customer's point of view. The fact that we had completely changed the architecture out from under them was invisible. I mean invisible.

Hendrie: Yeah.

Supnik: So--

Hendrie: So it didn't matter--

Supnik: It didn't matter.

Hendrie: -- and the run-time structure was--

Supnik: Right.

Hendrie: -- their code at the compiler level, same old stuff.

Supnik: Right. And [we--] the team could change things under the covers to take advantage of the new technology that was evolving in the Intel space.

Hendrie: Yeah. Of course. Mm-hm.

Supnik: One of the last projects that I had kicked off was to completely rework the network stack. And what they did was they basically moved-- took advantage of APIs that were there, that customers relied on. They snipped below that, and they moved all the code below that that customers can't see or access over to the I/O processor and rewrote it in C to be native. And then they used the Intel hardware and the native operating system TCP stack. They had boosted throughput by a factor of 10 when I left, and they were just getting started.

Hendrie: Yeah.

Supnik: So it's not just that the CPU got faster, it's that the I/O gets faster, the networking gets faster.

Hendrie: Yeah.

Supnik: And [the-]the user's applications just run.

Hendrie: Mm-hm.

Supnik: The 36-bit machine was a little bit trickier because the-- one of the, if you remember the Burroughs architecture, it was-- it was lovely.

Hendrie: Yes.

Supnik: And it outlawed things like self-modifying code. It required that all code be signed by the compilers so that you couldn't insert a Trojan or anything like that. The 36-bit machine, the UNIVAC machine, was just a standard register-memory 36-bit machine. It looked a lot like a PDP-10. In fact, it looked a lot like a PDP-10 that had been through 25 more years of architectural ornamentation.

Hendrie: <laughs>

Supnik: So it was fairly-- it was fairly daunting to get its performance up to speed. And they took a much bigger leap of faith when they dropped out of the hardware business. In fact, the head engineer of the 36-bit group had written a paper to me saying, "You will never get to 600 MIPs. Here is the mathematical proof of why it's impossible."

Hendrie: Mm-hm.

Supnik: Right?

Hendrie: Okay.

Supnik: And, of course, when we got there he was very gracious and said, "Yeah. Well, I was wrong."

Hendrie: <laughs>

Supnik: So they had a tougher road, and they hadn't any place to start. They were not doing translation when I got there. So I put them in touch with a fellow named Maurice Marks, who had done most of the Alpha translators. 'Cause Alpha had translators not only for VAX code, but for SPARC code and MIPS code and this code and that code, and Muarice had done them. So I went up to, "Maurice, meet the 36-bit team, help them out." And he formed a new team, some veterans, but a lot of college hires, to go after this problem. And [they-] they worked at it and they worked at it and they worked at it. They took a little different approach. They used a standard compiler, a backend called [LLVM]-- the one that everybody uses now, the backend of Clang for-- Apple uses it. It's a very good code generator.

Hendrie: Mm-hm.

Supnik: And so what they were doing is they were compiling 36-bit instructions into intermediate language and then running it through a full-scale compiler backend.

Hendrie: Okay.

Supnik: And that was troublesome. It didn't seem to yield the improvements that they needed. And we were getting pretty desperate. We were coming up to an important announcement where we needed to conclusively demonstrate we were on the right track. We had a goal of 450 MIPS, and we were not there. And so the college students who, of course, don't know what can't be done, took this as a personal affront. And they just worked and worked, and they tried things, and if that didn't work, they tried something else, and they kept going, you know, working at it. And eventually they cracked it. And what they found--

Hendrie: Yeah, what they do?

Supnik: -- was they needed to make the footprint of the translator and emulator that was underneath it small enough that it fit in the primary cache of the Intel chip they were using, now 64 Kbytes. And once they did that-- once it was small enough, not only did the compiler and emulator run-- the just-in-time the compiler and emulator run fast enough, the resulting code ran fast enough because it wasn't thrashing the caches anymore.

Hendrie: Oh.

Supnik: So once again, the same lesson, the same going all the way back to God knows what, that small is better, small is faster, was proven. And as a result, they were able to lay out a roadmap with confidence to get to their goal of 600 MIPS, equaling the fastest performance they had ever built. So it's now 2014. And mentally I had set in my mind a date of July 2014 to retire, that being the 50th anniversary of my first job in computers.

Hendrie: <laughs> Okay.

Supnik: I figured 50 years was long enough.

Hendrie: Yeah.

Supnik: So I, in April of 2014, I went to see the CEO. I said, "Ed, I've got to tell you something," and he said, "I know what you're going to tell me. I've been dreading this moment for years. You're going to retire," and I said, "Yes." Said, "Look. I need some time. I need to make appropriate arrangements for transition in both the business and the technical side. Can you give me some time?" I said, "Sure." Because I'd go through fire for this guy. And we got to the finally late summer, early September. He calls me up, says, "Bob, I'm ready now. You can announce your retirement." So I tell my team, "I'm going to retire. I'm going to-- I know who my successor is. Everybody agrees he's the logical person. I'll stick around for four or five months and, you know--"

Gardner: Help out if I'm needed.

Supnik: Help out. Be a guru in residence. Otherwise, I'll be an individual contributor kibitzing with the translation teams. So everything looks really good. Three weeks later, the board fires the CEO. Worst damn decision a board has ever made. Now, I can understand why. The company had not grown. It had been unable to grow revenue, and if it couldn't grow revenue, well, what was its future?

Gardner: It was profitable though.

Supnik: It was profitable, and Ed knew how to keep the cost in line, and we had some promising new products that we thought could drive growth if given another year or two, but the board ran out of patience. They fired him. So at that point, I was glad to go. Because I didn't think they'd find anybody as good as him.

Gardner: Yeah. That you could relate to.

Supnik: Right.

Gardner: Certainly. Yeah.

Supnik: So I hung around for another five months, attended my last user meeting, attended a few other things, and retired at the end of February, 2015.

Gardner: Okay.

Supnik: For real.

Gardner: <laughs> For real, as opposed to not on purpose. <laughs>

Supnik: I had grandchildren. First grandchild, rather, and I also-- two. Two? No, one. So there were things at home that were new and different and wonderful, and my wife was no longer concerned about having me around all the time, and I had a hobby project. So now we must double back all the way to 1993 and talk about--

Gardner: To ’93 yes, yeah.

Supnik: And talk about SIMH.

Gardner: Okay.

Supnik: So all my life, all my career, I built simulators. You know, when I first joined Applied Data, I tried to write a simulator for the Interdata 4 on the PDP-7. Yeah.

Gardner: <laughs>

Supnik: I didn’t understand--

Gardner: Now, those were two ancient machines.

Supnik: I didn’t understand the I/O structure of the 4, and I never got it right, and the Interdata people just laughed at me.

Gardner: Okay.

Supnik: When I was--

Gardner: But you were a kid!

Supnik: Was a kid. When I was a consultant, one of the projects that ADR did for DEC was to write a 1401 simulator on the PDP-10.

Gardner: Oh, mm-hm.

Supnik: And Len Fehskens and I and another person did that. Later we wrote another 1401 simulator but for the 11, so that it would be more cost effective. To keep myself entertained, I wrote an 8 simulator on the 11 and it just, I always hacked around with it. Well, in 1993, when, late in '93 when the company was reorganized, and I was given this non-position of chief of technologist to this division that wanted to be in the PC business. I had time on my hands, and I realized that my programming skills had atrophied. I was still programming in VAX macrocode when I wanted to solve a problem. Not exactly what you would call a productive high-level language.

Gardner: <laughs>

Supnik: And I didn't think it was--

Gardner: Yes. But you could really squeeze it and tune it and it was a lot more fun.

Supnik: And I knew how it worked.

Gardner: Yeah. <laughs> Okay.

Supnik: And I didn't think there was going to be any more call for microprogrammers. Not with RISC machines.

Gardner: Right. <laughs> RISC instructions. <laughs>

Supnik: So needed a programming project. And Larry Stewart of CRL, said, "You know, the history of computing is disappearing. Why don't you take your interest in simulators and write some simulators that will allow us to preserve the software base of all of these interesting machines? For educational purposes," and then he had to tactfully, "And why don't you write it in something people might be able to compile, like C?"

Gardner: <laughs>

Supnik: So I taught myself C, and I started writing simulators, and basically I used the design of the PDP-10 simulation system MIMIC, that we had designed back in ’69 and ’70. Just transported, you know, ported it to C or transposed it to C, thus proving that C is really assembly language with different syntax, which is something everybody’s always known, and so I started just knocking the machines off. I wrote an 8, I wrote an 11, I wrote a Nova, I wrote an 18-bit simulator. I wrote a-- got the Interdata right this time, and while as long as I was at DEC, I was able to write licenses to myself saying, “This is now free to be used for non-commercial purposes,” so…

<laughter>

Supnik: We got--

Gardner: Well, and you’re writing on DEC machines. Yes, yeah.

Supnik: Right. Well, the license for the simulators was always a precursor of open source. It was always, the copyrights were always written as MIT style copyrights. It was getting the--

Gardner: What’s that mean?

Supnik: Mean, “Do any damn thing you want with it and don’t sue me.” <laughs> That’s really, really what the MIT X11 license says, is, “Preserve my copyright, do anything you please. Don’t sue me.”

Gardner: Right.

Supnik: [The interesting part] It was really getting a license for the software. Being able to have piece of paper that said, “You are free to use DEC’s old commercial software for the PDP-8 without charge for non-commercial purposes.”

Gardner: Right.

Supnik: “You are free to use DEC’s old PDP-10 software.”

Gardner: Yes. Yeah. That’s really valuable.

Supnik: Right. And as a VP, I could sign these things. It was great.

Gardner: Okay.

Supnik: And once-- so I ripped through almost all the machines I've used in my career by 2000, and started writing a VAX simulator. Wrote a VAX simulator because DEC had a free-for-hobbyists program for VMS. It was very nice, and HP continued it, and then with the Internet, I started getting some collaborators. People wanted to work on this piece, or they had this machine they wanted to do. Somebody wanted to do the HP 16-bit machines. The internet became the way to solve some of the problems of, "Where is the software now? How do you find it?" The PDP-- the 18-bit machines for DEC, which were pretty scarce. Finding the software required a nine-year hunt that spanned four continents.

Gardner: Wow.

Supnik: One piece was recovered in France, another piece was recovered in Australia, another piece was recovered here and there, and we just went from strength to strength. Trying to move backwards in history, which we were trying to, I was trying to make sure that we captured all of the key time-sharing systems, CTSS, the SDS 940. TOPS-10 and 20, and so on. We wanted to capture significant milestones,like the history of UNIX insofar it was as it had been preserved, and the availability of the simulators drove people to go look for things. So when originally we were starting, people thought that almost all the early versions of UNIX had been lost, but as a result of having simulators, we recovered the first 32-bit version of UNIX, which was written for the Interdata machines, and got it running. First, we pushed working UNIX releases back through Version 6 to Version 5, then to Version 3.

Gardner: What was Version 3 written on?

Supnik: It was all-- this was in C. Everything on the 11 was written in C.

Gardner: Oh, so yeah, you haven't gone back to the 7.

Supnik: Yes, we have.

Gardner: Oh. Okay. You're getting to there. Okay. I'm jumping ahead in the story.

Supnik: You are. This was the triumph of last year. Everybody thought that the first version of UNIX, the one that ran on the 7, had been lost forever. Dennis Ritchie said so, and Ken Thompson said, "We don't have it," and then it turned out that some colleague of theirs had Xerox copies of the listings for at least much of the system in a notebook in his attic. Because of the publicity and the worldwide nature of the effort, he got hooked up with the people who run the UNIX Historical Archive in Australia, and said, "Yeah. I'll give you a copy. It's okay. I'll just take it to the Kinko's and send it to you." So now they had this paper listing, several generations old. [It] resisted OCR. So they just divided and conquered. They said, "You know, we're going to get-- there are 18 people."

Gardner: Ten people who want to work on it or whatever.

Supnik: Right. "And we're going to type it in, and we're going to get it running," and the guy who's in charge of this was named Warren Toomey. Never worked on a PDP-7 before. But he knew-- I know SIMH, so he called me via the internet. He's in Australia. I actually met him later. I mean, I'd worked with him for 10 years and I finally met him. He said, "Can you help? Look at this code. We have no idea what it's running on." So I read through the code, the first PDP-7 assembly language I've looked through since 1967.

Gardner: Yeah. <laughs>

Supnik: And I found, for example, the disk handler. Looked at the instructions and the algorithm that was being used, and thought, "I've seen this before. This code is constructing decimal track and sector addresses. I know I've seen a machine like that, a disk like that," and sure enough, a disk like that had existed on the PDP-9. So I made a leap of faith and said, "Well, they must've somehow done a controller for the 7 for the same drive, [the] same fixed-head disk."

Gardner: Yeah.

Supnik: A whopping one-megaword.

Gardner: Okay. It was a fixed-head disk. Yeah, yeah.

Supnik: It was a fixed-head disk.

Gardner: Yeah.

Supnik: And--

Gardner: I would've thought it'd have been drum in that era, but maybe it wasn't.

Supnik: No. It was an old-- believe it or not, it was an old Burroughs disk.

Gardner: <laughs> Okay.

Supnik: And the clue was that DEC called it the RB09 where “B,” stood for “Burroughs,” because that’s how they name things, rotating Burroughs.

Gardner: Yeah.

Supnik: And then another aficionado of old machines, Tim Litt, did the research to in fact find where in DEC’s ancient catalog of all known parts there actually was a description of this controller that had existed. There had been an RB09 controller for the 7 and it was the same controller, just with different interfaces. So creating that disk on the PDP-7 simulator was a hell of a lot easier than it must’ve been for the DEC engineers, because all I did was physically lift the module [from the PDP-9 simulator build] and drop it in[to the PDP-7 simulator build], and attach [it with] two lines of interface code, and voila, I had it.

Gardner: <laughs>

Supnik: And sure enough, they debugged worldwide using the internet, and they got UNIX running. UNIX Version 0.. Now, the listings are incomplete. They only go up-- they have the kernel and they have all of the utilities up through A through J. But the rest are missing. In particular, they’re missing the shell.

Gardner: <laughs> Yeah.

Supnik: And runoff and a few other things, so they--

Gardner: A few. <laughs> Just a few really interesting pieces.

Supnik: So they have created what they think those must’ve looked like. So you can now get a working UNIX V 0 system, it’s got a shell that corresponds sort of to the documentation that was available, and it’s obviously a reconstruction, not a reproduction. But--

Gardner: Hey, it’s pretty good.

Supnik: Hey, it’s pretty good.

Gardner: Wow.

Supnik: The simulators have been used to bring CTSS back to life. Another group used the core of the simulator, the control panel and the central structures, to build a GE 645 and do Multics.

Gardner: <laughs> You're kidding.

Supnik: No.

Gardner: <laughs> Whoo.

Supnik: Not only was there a complete HP 16-bit simulator, HP released all its 16-bit software for noncommercial use via the Computer Museum. So that's all available. I have the one and only RDOS license for non-commercial use for hobbyists in the world, courtesy of Tom West, for the Data General simulator.

Gardner: Okay.

Supnik: We have PDP-10 licenses, and you can go about as far back as you want. I mean, I think the oldest machine in the line is the 7094 and the 1401, and the 1620. So late '50s IBM machines.

Gardner: Yeah, that's really good.

Supnik: And then most modern is the VAX, and someone is working on an Alpha. It became a formal open-source project in 2008. It's on GitHub. Somebody else is now the chief editor, and it's really expanded. I mean, there's a Russian mainframe on there. There's a whole bunch of single-board computers and hobbyist computers on there. You can do VAX stations with graphics heads on your PC, so you can see what it was like to run a workstation in 1976. Or excuse me, 1985. It keeps me, it gives me a lot of entertainment. I'm still the person they turn to when there's a problem in what I would call the simulated hardware. That is, a lot of people who write simulators are software engineers. They sort of are working from functional specs. Sometimes that's not good enough.

Gardner: Yes. <laughs> Sometimes there's something that happens that--

Supnik: Yeah.

Gardner: --wasn't quite described in the functional [spec].

Supnik: Yeah. And the other thing is that because the simulator is much faster than any real machine that was ever built and can build much bigger configurations than were ever affordable we found operating system bugs.

Gardner: Ooh, yes.

Supnik: I filed--

Gardner: Because it was never stressed.

Supnik: Right.

Gardner: On--

Supnik: We filed-- we could've filed except there was nobody there to listen.

<laughter>

Supnik: The first bug in-- excuse me. Let me say. We generated the first patch for the 18-bit advanced software system in 30 years when we discovered that if you stacked up the maximum number of fixed-head disks it didn't work, because nobody could afford it.

Gardner: Yeah, so was it never tested?

Supnik: It was never tested.

Gardner: Yeah.

Supnik: So that's how I keep my hand in in terms of the computing technology. It's all back to the future now, and in addition I'm--

Gardner: That's fun though.

Supnik: --a volunteer IT administrator for the New England Quilt Museum. I help my fellow senior citizens.

Gardner: The New England what?

Supnik: Quilt Museum.

Gardner: Ooh.

Supnik: Right. My wife is a quilter and she's been the librarian there for--

Gardner: My sister-in-law's a quilter. That's very interesting.

Supnik: I think [dating] since the late '80s.

Gardner: Okay.

Supnik: And when they got computers, I was volunteered to be the IT person.

Gardner: <laughs> I know how that works. <laughs>

Supnik: It's fun. I'm--

Gardner: Didn't have a problem.

Supnik: --really enjoy. Keeps me up-to-date on current PCs, and so now I help fellow seniors in Carlisle with their PCs. I renovate discarded machines for charities and fix problems and generally hack around. That's why if you were to look around here, there's--

Gardner: Saw some interesting stuff.

Supnik: Yeah. There's like one, two, three, four computers on that bench, two more over here, and that's only because a bunch of them have been cleared out recently for use for good causes.

Gardner: Good.

Supnik: So…

Gardner: Okay.

Supnik: And that’s the end of the story as far as I know.

Gardner: Well, you have two more things.

Supnik: Okay.

Gardner: You had a correction that we haven’t gotten on tape.

Supnik: Oh, okay. Right. Yesterday I-- or two days ago or three days ago, I said that I first encountered computers in ’61, 1961. In fact, it was 1959 when I was 12 1/2. Dave Waks came to Thanksgiving dinner to court my oldest sister, and it being a long weekend with time to kill, he told me what he was doing. He was a math student at Cornell, and at Cornell, the computers were in the Math Department. They had a[n IBM] 650. David had written his first program in October of ’57 as a freshman. He was the first person to use the computer in assembly language instead of the Bell Labs Interpretive System or BLIS.

Gardner: Oh.

Supnik: So he had started using computers. He was hooked on them, and he got me hooked, and he also got me to believe that computers and math were connected. Now, I missed an important point, which was he was an applied mathematician.

Gardner: <laughs>

Supnik: Right. And applied math and computers sort of did go together.

Gardner: Right.

Supnik: When I went to MIT, it was a theoretical math department, and it sort of didn’t quite work the same way.

Gardner: Yes. Okay.

Supnik: And when I visited the Courant Institute in 1962, it was because he was a graduate student there while simultaneously working at Applied Data Research, which he had joined when he graduated in June of '61. So again, that reinforced the connection that computers belonged with math, and when you were asking who influenced me to go into math, I said it was my, you know, one of my high school teachers, it was also David, because I thought computers went with math.

Gardner: Yes.

Supnik: So that's the--

Gardner: Yeah, you get to use computers if you majored in Math. Yeah.

Supnik: That's what I thought.

Gardner: You get to play with computers.

Supnik: Turned out not to be the case.

Gardner: Yeah, that's what you thought. Well--

Supnik: At MIT.

Gardner: --all right. <laughs>

Supnik: And what was the second thing?

Gardner: The second thing was the - I wanted you to tell me a little bit about your work. I don't want to duplicate anything but you have to mention on your history your involvement in games.

Supnik: Oh, yes. Right.

Gardner: We can't leave that--

Supnik: Can't omit that, can we?

Gardner: --just totally blank.

Supnik: <laughs> Ah, yes. Okay. So it's 19-- we have to go to '79 now. I'm new to DEC, and DEC is being swept by a phenomenon called Adventure, Willy Crowther's first pioneering text adventure game.

Gardner: Right.

Supnik: Or maze game, or whatever you want to call them.

Gardner: Yeah.

Supnik: And the company goes crazy. All the engineers go crazy, and because it runs on a PDP-10 timesharing system, it's really bogging down the company's machines, which are few and expensive and precious.

Gardner: Mm-hm.

Supnik: So the company bans playing Adventure on the PDP-10, not only during the daytime, but even into the early evening, because engineers stay late.

Gardner: Yes.

Supnik: So there's no computer time available. In that day and age there were no home terminals, you know.

Gardner: Right.

Supnik: So I'm not going to put up with this. And the sources are a FORTRAN program. I know FORTRAN, and I have access to all the PDP-11s I can eat, because they're commonplace all over engineering. They're never in use during the day or the evening, so I think, "I'll get Adventure onto a PDP-11, and then I can play it whenever I want."

Gardner: Mm-hm. <laughs>

Supnik: And anybody can play it whenever they want.

Gardner: Self-interest. <laughs>

Supnik: Yeah. But the other thing is, I really wanted to know how it worked.

Gardner: Whoo.

Supnik: And I really wanted to know what the mechanisms were.

Gardner: In the game.

Supnik: In the game. Yeah.

Gardner: Yeah, yeah, yeah.

Supnik: So I was pretty expert in FORTRAN, so I took it apart and realized it was way the heck too big.

Gardner: Yeah. Well, yeah. You've read all the code and--

Supnik: Right.

Gardner: --then you study it and then you understood it.

Supnik: And so I started doing surgery, you know. The first thing was to take all of the text strings and move them into a disk file so that they weren't in the precious 28K words of code that a PDP-11 had, and then there were some controls that were appropriate for the big machine that we weren't going to need, like it had a feature for setting hours. Well, wouldn't need that.

Gardner: Yeah.

Supnik: It had a debug mode. Wouldn't need that. So I cut and cut and I got it to run with my goal, which was the smallest PDP-11 DEC made, an 11/03, the LSI-11, with two floppy disks, and--

Gardner: Whoo. <laughs>

Supnik: --turned that loose and was very happy with it, and that, it did help actually drive down the demand on the 10, so nobody was upset that I had done this. Now it's 19-- a little later. I'm sorry. I think I did Adventure maybe in '77, actually.

Gardner: Okay.

Supnik: Right when I got there. Yes, I'm still in Disk Engineering. So now it's a little later. It's coming into late '77 and early '78 and a new craze starts sweeping down.

Gardner: <laughs>

Supnik: And it's a program from MIT called Dungeon.

Gardner: Oooh. Yes.

Supnik: And Dungeon is far more sophisticated.

Gardner: <laughs> It's the death of my--

Supnik: And it's much bigger.

Gardner: Death of my oldest son. Yes. <laughs>

Supnik: It's complicated. The puzzles are fiendish. It's got a full syntax parser, so you can give it really complex commands instead of just subject verb, and I really, really, really want to know how it works. But we don't have source code.

Gardner: Okay. Do we know what it's written in?

Supnik: Well, it comes from MIT, from the Incompatible Timesharing System, ITS.

Gardner: Okay.

Supnik: And it turns out it's written in what's known as the model development language or MDL, which is a bastardized form of LISP, and-- but as I said, we don't have the source code.

Gardner: Right. Or a compiler probably. <laughs>

Supnik: Or a compiler or anything else.

Gardner: Or anything, yes.

Supnik: In fact, it's running on TOPS-20 in a compatibility mode TOPS-20 had that emulated ITS. So it wasn't even native code. Well, this, you know, I'm not going to put up with this.

<laughter>

Supnik: I mean, so Ted Hess, another ADR veteran who had joined DEC, also had connections at MIT, and he found some fragments of the source code for me to look at. I said, "Okay. We should try and get the whole thing," but by this time the developers were getting wary of it being propagated, and they had encrypted it. So Ted applies the combined computing power of the 10 group that he was in at night when nobody was using the machines and he cracks the encryption for me.

<laughter>

Supnik: No. I'm not even going to comment on the legality of all this.

Gardner: Yeah, yeah, yeah, okay. You don't need.

Supnik: The only thing I will say is there were no copyrights on the code at the time.

Gardner: Yes.

Supnik: So I have the code, I buy a MDL manual and, you know, what is this?

Gardner: What is this?

Supnik: And again, my goal was to get it on to an 11 so that people can play it.

Gardner: Yes. A machine that's ubiquitous at DEC.

Supnik: Right. So I have the listings, and I bring them home on a Monday, and it turns out it's the Monday of the Great Blizzard of '78.

Gardner: Okay.

Supnik: I left DEC early because they told us to go home. I put the listing in my briefcase and brought it home, and essentially over the next five days while we were all housebound, I worked out the mechanisms I would use to rewrite it in FORTRAN and pack it into a PDP-11.

Gardner: Wow. <laughs>

Supnik: So I then proceed to code up a skeleton of this. I don't want to tackle the syntax parser, but I get the, you know, quite a bit of the game running. You have to type in subject, verb, indirect object by hand, and then I asked Ted to arrange a meeting with these guys. The guys who had written it at MIT, the four authors, as they were called, and they come out from MIT. I show them this, and they're just flabbergasted. They're just flabbergasted that this mainframe monster program of theirs is running on this Tinkertoy machine.

Gardner: <laughs>

Supnik: And--

Gardner: Well, they had no need to even think of the word optimization.

Supnik: Right. So…

Gardner: <laughs>

Supnik: And of course the techniques you use in FORTRAN to map what's fundamentally a functional language are brutally ugly, and they're just, you know, they're as appalled as they are flabbergasted. But they say, "Okay. Go on ahead. Finish it." "This is-- see what happens." So I work at it. It's much more difficult this time because the program's much bigger. It ends up being not only having separate databases, but it's overlaid five levels deep and all kinds of problems, but again the same thing. I get it running on an LSI-11, with two floppy drives and show it to them, and a light bulb goes off in their heads. Which is, "If he can get this running on such a small machine--" because I was, this was a 56K byte PDP-11 with two-- what was it-- 256K byte floppy drives.

Gardner: Yeah.

Supnik: Then we can get it running on a personal computer. A hobbyist personal computer. You know, the ones that existed in the late '70s like the TSR-80 and the Atari and so forth.

Gardner: Yeah. Right.

Supnik: So [they,] they go off and they form Infocom, and Infocom proceeds to market Dungeon, now renamed Zork, in three pieces. They broke it up so that they could get it to fit without overlays, and then they go on to do a whole line of text adventure games, and it's very successful. Which means that the version I have, which is in DECUS [DEC Users' Group] and is available to anyone, is a bit of a threat.

Gardner: <laughs> Yes.

Supnik: So we--

Gardner: Yes. <laughs> Now there's money involved.

Supnik: Right.

Gardner: Oh, that'll--

Supnik: There's now money involved.

Gardner: That'll get people… <laughs>

Supnik: That-- yeah. Well, again, it was worked out fairly amicably. What they said is look….

Gardner: So what did you do? So what happened?

Supnik: They said, "Look. What's done is done. It's in DECUS. There's nothing we can do. What we want you to do is put our copyrights back on it, put a copyright notice on it that it cannot be used for commercial purposes, and re-up it to DECUS. Furthermore, encrypt the string database so that you can't just print the string database and see what all the responses should be and therefore what a lot of the secrets of the game are," and I already had a lot of obfuscation in there, not deliberately, but just because I needed the space. So for example, all the strings had been encoded in Radix-50, and not having a Radix-50 primitive in the FORTRAN compiler, they were octal numbers. So…

<laughter>

Supnik: You know.

Gardner: Yeah.

Supnik: It wasn't obvious--

Gardner: Somebody would have to do a little bit of work to--

Supnik: Would have to do a little bit of work, but that was what we did is their copyright went on it, the string database got encrypted, and that version lived in DECUS for a long time. Then I just stopped. I just wasn't going to have anything more to do with games. It was 1981. I was running the [Semiconductor] Advanced Development Group. I had real work to do and it's…

Gardner: Yeah. Okay.

Supnik: No free time. Well, in 1990, when Alpha was in the middle [of development], and I was just running the program office, I decided to revisit it and targeting the PDP-11 and all these hacks on it done to get it to fit, it was time to put it on the VAX, so I rewrote it for, both games, for the VAX. Got rid of all the craziness in terms of the overlay structures and so forth. Unpacked the strings, et cetera, and that was the final version, and if you ever see either Adventure or Dungeon on a Linux machine, it's those versions from 1990 translated through a FORTRAN to C compiler, translator, that you see.

Gardner: Ah, okay.

Supnik: So we were at SiCortex, and there's a single-board computer that we're using for a control processor. It's got a read-only version of Linux in it, and we fire it up, we do ls user/games, and right there is Adventure.

Gardner: <laughs>

Supnik: Fire that up and, yeah, it's the 350-point version of Adventure.

Gardner: <laughs>

Supnik: That I first got onto the PDP-11.

Gardner: Oh, my goodness.

Supnik: And then onto the VAX.

Gardner: I love it. <laughs>

Supnik: So…

Gardner: And it came back.

Supnik: It came back. I've never dealt with games since, but on the notion that the fruit doesn't fall far from the tree…

Gardner: Yeah.

Supnik: Both my sons are in the games industry.

Gardner: Is that right? They studied computers and then went into the games.

Supnik: And then they went into games. The oldest does flight simulators and the youngest does casual games… Younger does casual games.

Gardner: Wow. Okay.

Supnik: So…

Gardner: Very good.

END OF THE INTERVIEW

