Oral History of Linus Torvalds

Interviewed by:
Grady Booch

Edited by:
Dag Spicer

Recorded: July 27, 2008
Portland, Oregon

CHM Reference number: X4596.2008

Grady Booch: Here we are on July 25th [2008] in the lovely home of Linus Torvalds, interviewing him for the Computer History Museum. So, thank you very much for joining us here. Well, actually not for joining us, we're joining you. So, thank you for inviting us in. Let's begin at the very, very beginning. You were born where? Tell us a little about your childhood.

Linus Torvalds: I was born in Helsinki, 1969, and quite frankly, I don't remember much of my childhood. I'm told there are people who remember when they were small. My memory goes back to about when I was ten-years old and that's it, nothing very special, I'm afraid.

Booch: And your parents? Tell us a little bit about that.

Torvalds: My whole family is basically journalists. So my dad is a journalist, my mother is a journalist, my sister has become a journalist, my uncle on my father's side, my grandfather on my father's side. They're all journalists. So I'm actually the black sheep in the family, although there are a few scientists on my mother's side, so…

Booch: Scientists of what nature, what field where they?

Torvalds: My grandfather on my mother's side was a mathematician, statistics, and my uncle on that side is a physicist, so…

Booch: And indeed, claims are that you were either named after Linus from the Peanuts strip or Linus Pauling. Which is it?

Torvalds: They're both correct, apparently. I'm told that I have this dualistic nature of serious and not-so-serious.

Booch: It was stamped upon you at the very beginning, it sounds like.

Torvalds: Yeah, maybe.

Booch: Any siblings when you were growing up?

Torvalds: I've got a sister, full sister, and I've got three half brothers. So, they're all younger.

Booch: What was Helsinki like when you were growing up, any memories of being in Helsinki? Were you in the city proper or in a suburb or…

Torvalds: I was in the city proper. And I don't know if there's any specific memories, just the basic fact that I grew up basically walking everywhere because, I mean, European cities being like they are, much denser than here in the U.S., that was basically my mode of transportation all the way until I moved here. But it's a beautiful city, even if it's not maybe the most exciting place in the world.

Booch: Now your maternal grandfather was a bit of an influence upon you, as well, was he not? Tell us about-- Tell us about how he influenced you, because I understand there was something about a computer he bought.

Torvalds: Right. So, I think my maternal grandfather was, to some degree, my biggest influence. Not just because of the computer, but because even before the computer came to be, he was kind of, he was the scientist in the family that I was in, like, got to know fairly well. And he was teaching me math and that's actually how I then got to know the computer he bought for his own mathematical needs. He'd basically used it as a very glorified calculator. And I was helping him type in the programs, whether he actually needed that help or not, I don't actually know. And it may have been that it was just his way of getting me into computers, although it may also have been that he was just used to doing everything on paper. So I would take his programs that he wrote on paper and type them in. And I was something like 10 or 11 at the time.

Booch: So you wrote your first programs when you were 10 or 11?

Torvalds: Well, I didn't actually write my first programs that way, I mean I wrote his first programs. And then I started doing-- I'd never really modified his programs because statistics was never my forte, even later on when I actually had a minor in math. Statistics just goes over my head. But back then, I mean this was [the] early '80s, you'd get computer magazines and type in their programs. And at that point you start modifying them because they're some kind of simple game or something. So, yeah, maybe I started programming so that I actually wrote my own programs when I was 12, 13.

Booch: What was the computer he bought, and what was his full name, by the way?

Torvalds: His full name was Leo Törnqvist.

Booch: We better spell that one.

Torvalds: L-E-O, Leo is a real name in the U.S., too. Törnqvist, T-Ö, well, "O" with two dots on top, which doesn't even exist in English, R-N-Q-V-I-S-T. So, now I've lost my-- where was I?

Booch: His first computer, what was it?

Torvalds: Oh, it was…

Booch: Was it a Finnish computer?

Torvalds: No, it was a Commodore VIC-20, which didn't make it very big. I think most people probably remember the Commodore 64 or just a ‘64. That was the kind of follow-on successor, and much more successful.

Booch: Do you still have it?

Torvalds: No, no, I have to admit, I am not very-- I mean I don't get sentimental about my computers. The moment I've, see something newer and shinier, the old ones go out the window so fast. So I don't have my first, well my grandfather's first, computer. I don't have the computer I even started writing Linux on. They've all been given away.

Booch: But your first computer you bought in 1987, I understand. Do you remember that one? It was a Sinclair, apparently.

Torvalds: It would have probably been early in '87. It was a Sinclair QL, another completely unheard of machine, unless you happen to be British or you happen to just follow Sir Clive Sinclair's escapades into odd technology. And it actually ended up… it was a nice computer from a hardware standpoint even though the fact that it was odd again meant that I ended up writing all my own programs, because there were no programs that you could basically buy for it. So it was maybe a bad choice from a practical standpoint, but it actually was a good choice in the sense that-- yet another reason why I continued with the programming.

Booch: So you self-taught yourself programming, it sounds like.

Torvalds: Yes, I mean there were programming classes in high school at the time, but they were pretty basic and rudimentary.

Booch: So you were certainly doing things out of the norm of what your peers were doing. Were there other computer types that you associated with or were you sort of the guy out there?

Torvalds: Not really. I mean, there was usually somebody. There were other people with computers, most of them just played games. So there were a couple of people that I was together with and did actual programming, but mostly it was very solitary, probably that's just-- I mean me personally. There were computer clubs for the VIC-20, for example, where you could get together and do things like that. And I think I attended one meeting just because. I thought it was interesting but I wasn't social enough to actually keep up with that, so…

Booch: What kind of games did you make and play on that computer?

Torvalds: Most of the games were actually impacted not so much by game-play but what I was interested in making the hardware do. So, one of the games I remember best was actually one of my early games on the VIC-20, where I actually wrote this routine to just scroll the screen sideways, which was not that simple to do efficiently in graphics mode on a CPU that wasn't very powerful. But I had no real use for it, but I did it just because I was interested in the hardware and trying to make things go as fast as possible. And then I turned that into a sideways scrolling game, which was more of a, whatever-- it was a small submarine that had to avoid fish coming at you. It was not a good game, but that wasn't the point. So I basically never played the game, and my interest was always the computer itself and the program.

Booch: So you were doing mostly assembly language programming then?

Torvalds: At that point, I was doing almost exclusively assembly programming just because…

Booch: What was the microprocessor for that machine?

Torvalds: That was a 6502, so. And I was literally-- actually I wasn't even doing assembly programming, I was doing machine code because I would not, I did not have an assembler. I would do all my assembly on paper and write it as binary, just because. Again, I don't know, I didn't any money so I couldn't buy an assembler anyway, but be-- I didn't know any better. So it was like, I had to learn to do machine code programming by-- there were all these programs which had tables in them with just the tables of assembly language, which I thought that was how you did machine code. So that's how I ended up doing everything.

Booch: Do you remember opcodes to this day in your head?

Torvalds: I still remember some opcodes, not very many. I mean LDAs, HEX, A9 and that's like, that was the most common opcode, so that one I still remember, but that's pretty much the only one I think.

Booch: This is from around age 10, 11 or so, and then you eventually went to college. But let's fill in the era in between there. What was life like growing up in Helsinki, because you were actually in a Swedish speaking minority in the area? Did that have any impact upon you?

Torvalds: No. So Finland is a country that has officially two languages. Finnish is the majority language with about 95 percent of the population speaking it. But for historical reasons, because Finland used to be part of Sweden a long time ago, Swedish is still an official language and it's spoken by about five percent of the population, and I happen to be part of that five percent. So I grew up speaking-- and you can do it to different degrees. My wife, for example, is also part of the five percent that speaks Swedish, but she lived in an area where there were no Swedish speaking schools, most of the time, so she went to a Finnish school. So her Finnish is much stronger than mine is. I only used Finnish when I was going out to stores or later on at university, my friends all spoke Swedish. I went through Swedish school up until-- in fact even through university, some of the classes were in Swedish.

Booch: Your English is marvelous. When did you learn English?

Torvalds: Thanks. I actually started reading English when I was 11, and that was partly because the kind of literature I was interested in, science fiction, was much cheaper to buy in English and the selection was much better, too. So again, the fact that I had no money came up here as a reason for learning to read English. But actually, not until I moved to the U.S. did I ever speak English with anybody, so. And I remember moving here in '97, and for the first two weeks, I had a horrible headache because I had a really hard time understanding people. I had to really concentrate on the spoken language and that was fairly stressful. But one of the reasons my English is fairly natural is that Swedish is not that different a language, so, you can still-- people still recognize that, okay, where are you from, from my pronunciation. But it's my second language now. My Finnish is…

Booch: But here we are surrounded by books behind you, and so you did a lot of reading in science fiction in that age. What are some of your authors that you remember back then or stories that stick in your head?

Torvalds: My main authors back then were so-called hard science fiction, so it would be Asimov and Heinlein. Those were the two main ones. I mean my favorite book for the longest time was Heinlein's Martian story. Oh God, it is a long time since I've read that one.

Booch: The Martian Chronicles?

Torvalds: No, no. That one is Burroughs, isn't it?

Booch: I'm trying to think of it as well, too. We'll dig that one up [*Stranger in a Strange Land*].

Torvalds: Wow, it literally was my favorite book, whatever 15, 20 years ago and now I can't even remember the title. I'm sure it's somewhere behind there but I'm not going to even turn around to look.

Booch: What else did you like to do as a kid, were you a physical kind of kid?

Torvalds: No. I still am not a physical kind of kid. I hate sports with a passion, probably because my dad tried to make me do basketball and things like that, which I was not physically very tall anyway. And if you're not that interested in game sports, which I definitely wasn't, I never got into that whole thing. So I was very much a geek. I mean, I stayed inside and read or programmed, and that was what I did. I basically had no other interests in life. And quite frankly, that is still what I tend to do when I'm not doing anything else. It's still reading and sitting in front of the computer.

Booch: It's your hobby as well?

Torvalds: It is my hobby.

Booch: Were you influenced or being pressured to move into the area of journalism? Here you were reading a lot and you have all of this family around you that were journalists.

Torvalds: No, there was no pressure. I mean it's kind of a joke that everybody else is a journalist and I'm the black sheep. But there was certainly no pressure at all. I mean, coming from that kind of background, you had a strong pressure to speak well and have a strong language. And obviously the whole reading part was part of the whole culture in the family. But it was not, I will disown you if you don't become a journalist.

Booch: But would it be fair to say that that influence probably pushed you to think out of the box, would that be a fair statement? As a journalist, these were folks who probably were not just reporting on the world, but questioning the world too. Was that an element of it?

Torvalds: I don't know. I don't think the journalism part necessarily. I think that whole culture meant that, okay, communication was important and that books were considered important. But I'm not sure if it had any-- who knows? Trying to analyze yourself is not likely to be fruitful. You had to have somebody else kind of see the picture from the outside.

Booch: Then finally, university. Why did you choose to go where you went?

Torvalds: That was easy. I mean, basically I wanted to be a scientist. I mean that was what my grandfather had done to me, was if you're interested in computers and math and your biggest, like, idols are scientists, then it was very obvious that I wanted to go to university. I mean anything else would have been completely unacceptable in my family, too. And Helsinki University was the largest university in Finland and it was right there, so there was no real, even, I mean there was no choice. I could have gone to the technical university, which is much more-- it's not actually in Helsinki. It's just outside of Helsinki, which is much more practically oriented and is more of an electrical engineering place. And to some degree, that would probably have suited me better since I was very practically oriented, but at the same time I really enjoyed the theoretical side, too.

Booch: Theoretical side of what kind of science, because you mentioned math and physics.

Torvalds: I did most math and computer science. But I mean, the computer science was computer science, it was not programming. So most of the courses at the university would be things like complexity analysis, which things like that where you don't really even necessarily do a lot of programming, if any at all, but you discuss the problems, [at] more of an algorithmic level. I don't know how much immediate practical use that ever really had, but the mindset of thinking about the problem at a higher level, as I'm convinced, is pretty useful. Since I had the low level, that part the university didn't really help with or try to teach at all.

Booch: Any professors come to mind that really were influential for you?

Torvalds: No, not really. I think at that point, in fact university-- I was pretty self-sufficient and I've--maybe not fully-formed but certainly not longer my formative years. And I think that if, when it comes to teachers, aside from my grandfather, it would have been my math teacher in high school who was more influential there.

Booch: What level of math was he bringing about, calculus?

Torvalds: Math, I think math in high school in Finland is already at a higher level than it probably is here, unless you take some advanced math classes in the U.S. But just as an example, he basically wrote the book that everybody else used in high school. And the first lecture we ever had from him, he said, okay, I wrote this book. I will not use it. And he basically acted the way math ended up being taught at a university level, where he'd give out notes and you weren't required to do your own notes because he felt that that was a waste of your time and his. And it was much better that he gave everybody his notes and you [were] supposed to listen and think about it instead of trying to write it down. In fact, when I went to university, I was wavering between doing math and computer science as my major, just because I actually thought I liked math better. But at the same time, computers was kind of what I did as a hobby, and there was also a very obvious path from computers to actually getting employed. So I ended up doing computers as a major, but math was a very strong minor for me.

Booth: Did you imagine at that time that you'd probably just go into some business in the country or did you have a grand plan?

Torvalds: I did not really have a grand plan. I mean for the longest time my grand plan was to stay at the university, just because I thought I'd-- kind of getting a Ph.D. and working upwards and doing my-- eventually being a professor at university was kind of my grand plan for the longest time until I realized I really, really, really hate writing papers. And let's face it, if you can't write papers, there's no point, you should not stay at university. So I ended up being at the university for awhile, just because I enjoyed it a lot and it was also the social scene. But I actually decided I don’t need the Ph.D. and end up with just a Masters.

Booch: So you went there for six years total, then?

Torvalds: I think it was more. Part of it was, in Finland you have to go-- well, you don’t' have to but it is customary to do a year of army. So there was one year in between that I was in the Finnish army.

Booch: So why am I having a hard time envisioning you in the army?

Torvalds: I don't know. It was not I think a great year for me or my superiors in the army, but it's part of the culture. And it means that you can go out with any Finnish male for a beer and you are guaranteed to have something you can talk about. So, it's actually a socially very-- very good.

Booch: What did you do for the year in the army?

Torvalds: Half of it you just run around within the forests in Finland and you, you're just generally miserable. And then at one point, you can choose whether you want to actually become a-- just stay a grunt, which gets you out much earlier, or if you want to go through officer's school, which means that you have to stay three more months, actually two more at that point.

Booch: Which path did you take?

Torvalds: I took the long path just because-- I mean, partly it's ego, partly it's you've just-- having run around enough in the forests in Finland, I have to say that turned me off to the whole camping thing too, because, yeah, we were-- I was, probably three months of that year, I was in a tent somewhere in the middle of nowhere.

Booch: I did notice a tent on the way up here.

Torvalds: That's for the kids.

Booch: I presume that's not yours.

Torvalds: No, it is our tent, but the kids use that one. Yeah, I wanted to do something.

Booch: Were you commissioned as an officer, then? Did you make it to being an officer?

Torvalds: I am a, what is it, Second Lieutenant in the Reserve in the Finnish army.

Booch: Wow, impressive.

Torvalds: Well, it's not really impressive, when you realize that everybody goes to the army in Finland. It's not like I did Special Forces or anything like that. But it…

Booch: We'll leave that to your wife who's a karate expert.

Torvalds: Yes. It taught me how to sit around and freeze my butt off. That's basically what the army…

Booch: That's the lesson.

Torvalds: That is the lesson from the army, as far as I can tell.

Booch: So after the hiatus in the army, along the way in the university, you learned C finally, as well, too. What was that like? It was your first higher-level programming language, I presume?

Torvalds: It was my first higher, I mean, unless you call BASIC a higher-level language, which I actually don't think BASIC is that bad. People just like saying bad things about BASIC. C and particularly Unix, it was actually the same course that did both, was like a big revelation for me, not just for the whole programming language side but also because it was a completely new way of doing development and you had-- when you did something wrong, the machine wouldn't crash, right? And you'd be able to use a debugger and actually be fairly safe in knowing that you can't do anything really bad.

Booch: What were you programming on in university? What machines did you have back then?

Torvalds: I think the original one-- well when I started out it wasn't actually C, it was Pascal on a, I'd like to say a MicroVax. I think Pascal on VMS. And unlike the whole C and Unix part, that was not a revelation at all. Pascal on VMS was just a sad state of affairs. I hated VMS, I hated Pascal and it was not a good thing for me. I thought the extended Pascals that you could get on PCs were much more pleasant and much more useful, and had all the extensions that actually made the language useful. But then you had the whole crashing and burning thing again, right? So it was kind of a mixed bag. But once we went over to C and Unix, it was originally I think a MicroVAX or an ULTRIX or something like this. And I think that had something to do with the whole VMS thing, that it came from the same vendor. And then later on, it was Sun and SunOS.

Booch: I remember around that time, weren't they distributing the source to Unix to universities? Were you ever able to dig into and look?

Torvalds: They did. Well Helsinki University had done operating systems during the '60s and '70s. And by the time I went to university, operating systems weren't even considered-- A, Helsinki University wasn't that practical to begin with, right? So it wasn't that much hands-on, but operating systems were considered to be okay. It's done, we'll teach you about the basic synchronization primitives, and by that, they meant the really basic ones, like Peterson's algorithms and whatever, I mean things that nobody actually uses because the hardware doesn’t work that way anymore. But there was no OS-level work. So Helsinki University did not have-- it may be that the University of Technology had a license.

Booch: So the university sort of viewed it as a closed problem, sounds like?

Torvalds: Yeah. And people really thought it was pretty odd that Linux came out of Helsinki University because-- including the people at Helsinki University, it did not kind of fit in the whole-- it was not what they were teaching, it was not what people were kind of expected to do. Helsinki University was a great place, but it didn't have the culture of doing operating system research. But actually, I'm happy to say Linux changed that to some degree. I know they've had classes in Linux since, so…

Booch: Now this may have been before or after your army experience? Around this time, you bought your first IBM PC compatible?

Torvalds: Yes.

Booch: What machine was that, do you remember?

Torvalds: That was a white box, just the cheapest-- there's a theme here, the cheapest 32-bit PC I could find. So I had grown up with the 6502 and then the Sinclair QL was a Motorola 68000-based machine. So I always looked down on, I thought the Z80 was a horrible CPU. I thought the 8086 was just a Z80 on steroids, not worth bothering with. But then the [Intel] 80386 came out and that kind of changed the whole landscape.

Booch: How did it change it for you?

Torvalds: Suddenly, PCs didn't have this uninteresting hardware anymore. Suddenly PCs had the most interesting CPU in the market, as far as I was concerned. I turned out to be right. But it was really, you could already tell by then that the [Motorola] 68000 line was kind of, it was not really advancing. It was not getting faster. It was really just standing still. And suddenly you could get these fairly cheap machines that performed very well and had all these interesting things. I mean, one of things people hate about the PC is its baroque instruction set. And it really is like very ASCII, and that was what I found very interesting. I found the fact that, okay, you can actually do task switching in hardware. Hey, that's kind of interesting to me. That was one of things that made me want to buy this thing and play with all the features it had that I had not had access to in my previous machines. So basically, it was Christmas money and whatever, stipends from the school, bonuses, things like that, and I scrounged up enough money to buy my first PC. And I was never interested in running DOS in Windows, I was always interested in-- actually the CPU itself was the most exciting part. And then I knew from university by then that I wanted to run Unix on it, eventually.

Booch: Just trying to remember, this is '88 something? How much would that have cost? Give me some sense for how much you had to scrounge together.

Torvalds: Well, I think it was…

Booch: ‘91, actually was that date, I think.

Torvalds: Yeah, it was. I got the machine January 2, 1991 I think. Maybe it was early. It was like…

Booch: Definitely post-Christmas.

Torvalds: After Christmas break, that why I know it was very early that year. I'd like to say it was 10,000 Fin[nish] Marks, roughly. And at that point, it was about 5 Fin Marks to the dollar. So…

Booch: How much memory did this beast have?

Torvalds: It had 4MB of RAM, and it did not have a floating point accelerator at that point. I ended up buying one later, just-- not because I cared about floating point, but because I wanted to support the thing in Linux. I mean, it was by far the fastest machine I'd ever had. I'd like to say that it was a DX2 running at 66-MHz, but now that I'm-- no, that was a 486. So that was the next machine, so it must have been like a 33 MHz 386 or something like that.

Booch: So is this the machine where you first began working on Linux?

Torvalds: Yes, yes.

Booch: So tell me what inspired you to say: 'I want to write a new operating system for myself.'

Torvalds: Well, nothing inspired me to say that. That was not what I actually started out doing. I ordered MINIX off the, actually from a bookstore, because it actually came with, well, it was kind of the companion to Andrew Tanenbaum's book. So you [wouldn't] actually go to a computer store to get MINIX. You would actually go to a bookstore and order it. That took awhile to just arrive, and I was playing around with the machine until it came. But even after I had Unix on the machine, remember what I actually wanted to do was play with the CPU itself, so what I ended up doing, I just wrote my own programs to boot off the floppy, with no OS at all because having the OS in between me and the CPU was against the whole point of the exercise. So I wrote this boot loader. I wrote the program to get it to protected mode, which is where you can actually start using all these features that the CPU had. And that took me a long time just because, hey, this is a new architecture for me. Everything for me was completely new. But I got it working and then I had MINIX for my real work. But I wasn't actually doing anything there except connecting to the university computers. And so I took my small boot loader and-- MINIX actually had horrible terminal emulation packages. It was sad, it was really disgusting. So that kind of made me think, okay, I can write my own terminal emulation package. And since I'd already done this thing that switched tasks, which was what I wanted to do, I made my small task switching thing do the-- read from the keyboard and right to the serial line to talk to my modem, and the other tasks would read from the serial line and write to the screen, which is the simplest way to do terminal emulation program, and it's how you actually do it under Unix. You'd do it as two different processes. And I did just that, I just did it the hard way, right. I basically had this toy program that was useless for anything. I couldn't even download anything because I didn't have an operating system. So I couldn't write to disk because there was no file system, there was nothing. So the only thing I could do was basically read Internet news, which was what I did for awhile. And then at some point, I just wanted to download things and realized that I actually had to extend my small program to the point where it kind of was an operating system.

Booch: You use a couple of phrases I wanted to key upon. When you say you connected to the university's computers, of course connecting back then is very different than connecting now. And you mentioned the Internet. The Internet back then was very different than it is now. Tell us about it.

Torvalds: So, the connecting part was over a MODEM and dial-up.

Booch: Do you remember the speed of your MODEM?

Torvalds: I started at 300 baud, so it wasn't really horrible, but I pretty quickly upgraded to 2400. I stayed with 2400 for awhile. I mean the Internet was the same. It wasn't graphical, but I mean…

Booch: So you were on the newsgroups and such…

Torvalds: Yeah, I was on the newsgroups.

Booch: What newsgroups did you hang out in, primarily?

Torvalds: I followed comp.arch. [a computer architecture newsgroup – Ed.] One my time-waster newsgroups was alt.fan.warlord, which was this-- I think it may still exist as a newsgroup, although I guess nobody sees the alt category anymore because it's all child pornography, whatever. But back then, alt.fan.warlord was this newsgroup where you made fun of people's signatures, right. I don't think signature files really exist anymore, but back then, there were people who had like these 40 line signature files with really ugly, butt-ugly ASCII graphics in them like, and the whole newsgroup was just about finding people with the most outrageous signature files and scoring them by how bad they were.

Booch: Did you have an outrageous signature file?

Torvalds: No, no, no, I never had one. I think being, seeing everybody laugh at other people's signature files made sure that you kept to-- the official limits was, if it's more than four lines long, it's in bad taste. But really, if it was more than one line, it was pretty debatable. It was all just a time-wasting thing for me. I mean, I…

Booch: You were also into a bit of computer gaming back then, as well, weren't you? *Prince of Persia*, if I'm not mistaken?

Torvalds: Well, that was actually before I-- so before my machine was useful, before I had MINIX on it, there was nothing I really could do with it. I was doing some programming, but it was in DOS and it was not-- so for a couple of months there, my computer was basically this doorstop waiting for Unix to arrive in the form of MINIX. And during that time, I played *Prince of Persia* and…

Booch: Were you very good?

Torvalds: Not really, no. I was okay. I mean if you play anything for two months, you're good, I mean. But I was never that interested in gaming. But it was actually one of the few times in my life when I played a game for more than a week or two. So…

Booch: Let's take a break here, we need to switch tapes. We'll catch our breath and from there, we'll move…

END OF TAPE 1 / BEGINNING OF TAPE 2

Booch: …interesting juncture here, because, let's see, January '91 you bought your PC and you started in earnest working upon it. And as you were telling me, working from the terminal emulation program wanting more and more. So talk to me about how you began to start writing an operating system? What led you to that? What form did it take in those early days?

Torvalds: Well, once I got to realize that I had to expand in my terminal emulation package, all of that had been bringing in an assembler initially. And I realized, okay, it was fine for getting to really know the low-level CPU stuff and it was fine for the small things I needed to do for the terminator emulation package. But if I wanted to expand it and actually do a file system and do things like save and uploading to the university, I needed to just turn things around a bit and not work at that lower level. And I kept most of the low-level code, but I started working on trying to-- reorganizing the things so that it was-- acted more like a real operating system. And I had obviously been reading operating system books anyway, since that's how I knew about MINIX in the first place. So…

Booch: So there was Andrew's book, and were there others?

Torvalds: That was Andrew's book. There was Bach what is his first name [Maurice], I forget, *Design of the Unix Operating System*, or Design and-- well, was it? Yeah, something like that. It's a long time ago. My mind is going, Dave. Those were the two main ones. Tanenbaum's book about MINIX, which was kind of nice for the concepts, and then Bach's book about Unix, which was more about the practical implementation of Unix and listed the actual system calls. In fact, at the end Bach's book is a list of system calls and my copy of that has small stars next to them because I started going through the system calls. And as I implemented one I would put a star next to it and say, okay, done, what's the next one?

And it wasn’t really organized, but it was, like, I was literally looking at that book to see, okay, what do I need to write so that I can then move my terminal emulation code up a layer so that it would actually be a real process and it could do read-writes and actually open files and things like that. And that took a few months. That was also actually when I started talking to people about this project, because I had the book, the Bach book. But that was actually a fairly old book even at the time. And what I wanted to do, I wanted to be more standards-oriented with more modern standards and POSIX in particular. And I found some of it-- I mean I may have used the Bach book as the kind of basis. But then, for example, the Sun machines at the university were much more modern and had more of the modern interfaces. So the amount of [man, manual] pages there helped me some. But I was starting to ask around. So where's the POSIX standard for what are your system calls really supposed to do, instead of having to try to figure it out from really a book or from somebody else's man pages that are-- I mean, that is probably better, but it was still not what I was really looking for. And I was then approached by one of the people at Helsinki University of Technology who said that, "I'm sorry, but the POSIX standard is not actually available for free. You have to pay…," like, I forget what it was. But it was 50 bucks or something like that. And the theme continues. I mean, I'd already spent all my money on getting the computer. I thought 50 bucks was outrageous. Right? So I had to make due without it. At a later date people actually ended up sending me a copy, right. So now I do have a paper copy of it. And obviously POSIX actually made some of the standards, most of them available for free on the Internet, but that was-- by then it was already way too late. But what happened was, Artie Lencke [ph?] who was the person who told me hey, there's no standard, also said, "But hey, if you get far enough I maintain…," he had this interest in operating systems which was why he emailed me. And he maintained one of the FTP sites at the University of Technology and said-- he ended up saying that, "When you're far enough along, I have a place for you where you can upload this program." And so during summer of '91 I was basically working towards something that I knew was going to be an operating system, and my target was basically being at about the level of MINIX, except with a much better terminal emulation package. That was literally my target at the time. And I had been talking to some people just because I'd been looking for information. But I wasn't ready to release yet. And at the time my plan for the release name was to call it FREAX. Linux was kind of always the code name, and let's face it, I thought it was a bit too egoistical to actually call it Linux.

Booch: Did you first come up with that name, or did your colleague?

Torvalds: I forget. It might have been my own name that I kind of used. It may have been Otti, it might have been Lars who was another person who was kind of aware of what I was doing who called it Linux. But I didn't want to release it under that name. So I wanted to release it under FREAX because it was a free Unix, but also actually written like a F-R-E-A-X because it was kind of a freak thing. It wasn't to be taken serious. But I think in September or something like that I actually had my first version where the definition of the first version was-- I could actually run something under it. It couldn't do a whole lot, but it actually ran real programs, mainly a shell.

Booch: Do you remember your first email address back then?

Torvalds: My first email address was actually on the VMS system, and that was linus@helsinki.fi or something like that. But then they realized they'd made a mistake and they couldn't actually use first names, even unusual first names that nobody else had. So my email address at the time was Torvalds. It had been changed over, and I kept it that way since, even afterwards when people say now you can call yourself Linus, I'm now so used to be Torvalds at; it was torvalds@helsinki.fi. But I think the machine I was mostly on was ‘clob.’ So if you'd seen my old emails they'd usually torvalds@clob.helsinki.fi. And I think that's the one I used when I-- so now, actually the first announcement wasn't even public. It was sent out to just the few people I'd talked to saying, "Hey, it's there." I wasn't that proud of the first version. It ran a few programs, but it wasn't actually really usable for anybody, and it had some known issues. So the first version was not really ever made publicly announced.

Booch: There was the August announcement in '91. Was that that announcement or was that a subsequent one?

Torvalds: I think the August announcement was really a pre-announcement. That was the announcement that I was working on. I hadn't really made a release yet, if I remember correctly. I mean, this is going back some years. And I was telling people that hey, this is what I'm working on. I'm trying to basically do a clone of MINIX, which was the target at the time. And I was asking for input. I was really asking people-- at that point it was like, it actually did a lot of what-- I could see that it would do everything

I wanted it to do at that point. I mean, this was very naïve looking back, but I had really low expectations of what the whole system was supposed to do. But I also wanted to continue my project because I used to always do programming. And I wanted to have motivation to continue this project, and I could kind of see that it was nearing its end. So the first announcement was actually asking for people to comment on what they wanted to see from a MINIX clone.

Booch: What kind of reaction did you get from that announcement?

Torvalds: It wasn't a huge reaction at first. The only people who would be-- hadn't even seen it since it was made to the MINIX newsgroups where people who were kind of interested in operating systems to begin with were MINIX oriented, usually at the universities and still wanted to maybe see something else. And most people probably just thought I was full if it. And so you can't expect to take somebody seriously who says, "Hey, I'm writing my own OS." I mean, crazy people do that. And so I had a few people contact me. And there was some thread-- in fact I think most of the reaction to that it was-- at the same time I'd also made this silly hack to make my finger information change automatically. And I think that may have been the same announcement where I told people in a PS, "PS, try to finger me…," nobody uses finger these days anymore. But that was a way of getting information. And I have this magic named pipe thing that makes it give different things every time you did it to me, and I thought that was cool. And apparently a lot more people thought that was cool in my operating system, because I have this pretty strong memory of that one getting much more attention than the original announcement itself, you know. So…

Booch: So things became reasonably stable in the September timeframe, and then October you released the first official version, I understand.

Torvalds: Actually, it was a huge relief to actually get something working. And I think that's why doing operating systems is so crazy, because there's this-- I mean for six-plus months it was completely useless. I mean, it wasn't very useful even in October. But before that it was-- it did nothing at all. So for six months I worked alone on this thing pretty much full time. And it was not usable for anybody. So there's this huge initial step you have to take to get it into any shape at all. But then in September, October what happened was some of these small programs started working. And when something starts working suddenly it's not that hard anymore. Suddenly it's all incremental. I mean people didn't actually start helping me until like December, January of '92 maybe. But even my own productivity just shot up when I could just start compiling programs and saying, "Okay, that doesn't work." And the way I actually approached that was when-- I would take random open-source programs off that net. And when they didn't work, instead of porting the programs to Linux, I would see that as okay, this is a deficiency in Linux. So I usually would try to configure it to look like SunOS, which was [what] I was most comfortable with, because that was the university OS. Sometimes it's MINIX, and just say, "Hey, I will make this work. I will take this unmodified source code from the Internet and I will make sure that I extend on my capabilities so that it works, instead of changing it so that it works despite my lack of capabilities."

Booch: So sort of a random selection of this was an interesting program and…

Torvalds: I mean, it wasn't completely random. Most of them were programs that I used under MINIX. So it was some very basic-- I mean there's all the really basic Unix tools, the shells, the compiler itself was actually a big step when I could compile the compiler itself under Linux. Actually the first step was compile Linux itself, because Linux was a smaller project than trying to recompile the compiler itself. But there were all these basic tools that I tried to basically make Linux good enough to run.

Booch: So back then how big was Linux source line to code? Can you give me an estimate?

Torvalds: It was small. It was small. I mean, the very first release was something like 10,000 lines. And that sounds really sad in the sense that that was something I'd worked on for six months, and 10,000 new lines over six months is not a lot of code. But it comes back to the whole thing where it was so-- I mean, it was so much harder when nothing worked, so. Lines got written many, many, many times. I mean I think it doubled within months, I mean just a couple of months and it was double, and even before people started being involved. And now obviously I mean 10,000 lines, that's hardly-- you can see the current kernel is whatever, seven million lines or something.

Booch: The kernel is seven million? Wow.

Torvalds: Well, most of it is device drivers or Sun architectural support. But yeah, it's something like seven million lines of code.

Booch: So a bit of a leap.

Torvalds: It is, just being Sun…

Booch: And of course the original 10,000 lines is all beautifully documented and completely clear.

Torvalds: You know, I never believed in that whole documentation thing. Comments is for a code that you can't understand. And instead of adding comments you should just rewrite the code. That's my theory.

Booch: So you're a fan of literate programming in a sense.

Torvalds: Yes, in a sense, but not the sense that most people would use that term for, yes. I'm not. We do have a lot more comments now, obviously. I mean the situation we're in now is we're different from the situation back then. But it used to be that the whole 10,000 lines probably did not have more than 100 lines of comments because I'm not much…

Booch: Well, about the only eyes on it at that time were your two eyes.

Torvalds: Right. And I really do think that I like my functions small and obvious. And if you really need to comment, then you're doing something wrong if they're your own functions.

Booch: Is that a style that you morph to as you had programmed over the years, or what led you to that philosophy?

Torvalds: I expect it actually is because I started doing assembly language not as assembly language, but as machine code. And if you can't make it small and obvious, you can't maintain it. I mean, you'd better get it right the first time kind of thing. So I've always wanted to make-- my functions are small and I'm very happy to have a compiler that will then inline all those functions and make it more efficient than it would be if it actually stayed that way. But…

Booch: I'm thinking [this] is around the time of Charles Simonyi's Hungarian notation as well, too. Do you have a particular naming style? What was your reaction to that?

Torvalds: I mean, the Hungarian notation thing was like I was going, "Who would ever want to do that? You're doing something wrong. You're trying to tag your variable with their types. If you have so many variables that you can't even remember their types, you're doing something wrong." I mean, I have strong opinions about a lot of things, and one of my strong opinions is people can't really remember more than about six or seven things in short-term memory. So if your function has more than six or seven variables, you have already lost. Right? But if you only had six or seven variables in your function, then Hungarian notation just does not add anything at all, and it guarantees that you cannot change the type ever again. Because if you ever change the type now you have to rename it. I mean Hungarian notation obviously died a painful death. I don't think it's used anywhere.

Booch: [It] still exists in the [Microsoft] Word code base.

Torvalds: I guess it still exists in post code that was Windows back then, but I'm a big believer in having compilers that do static-type checking. And I mean really static-type checking. I think runtime-type checking is a horrible disease, because you actually have to run the code to find the type check errors, which nobody really runs the code. I mean, if it compiles, you ship it. Right? It's sad, but I'd rather have a high level of coverage from the compiler.

Booch: Sure. So to leap forward a little bit, you're not a fan, therefore, of dynamically-type languages or loosely-type languages?

Torvalds: No, I detest dynamically-type languages where you-- and people say that C is a loosely-type language, but actually if you write C the right way, which includes requiring a fairly modern compiler, where modern means in the last 20 years, right, C is a very strongly-type language.

Booch: Is it your favorite language?

Torvalds: It is still my favorite language. I mean, it probably comes from what I do. I do system programming. For system programming I have not seen anything better. For other things I would not want to do like a GUI [Graphical User Interface] thing in C. There's no question about that.

Booch: How did other people get involved in kernel work? Did they volunteer themselves? Did you seek their support?

Torvalds: They always volunteered. They still do. I mean, I wouldn't even want to work with people who don't, I mean, feel passionately about what they do because then, I mean, searching for people to do something, that just doesn't work. You want them to maintain it for years and years. So yeah, it started out slow and on a very small scale. But it was a very natural progression. Mostly just people said, hey, sometimes they said this is wrong, right, and sent a patch to fix it. More often they just said this is wrong and didn't send a patch, but they at least told me why it was wrong, which is almost equally useful. I mean, especially back when it was so small, just pointing out bugs was-- or quite often it was things I had literally misunderstood, right, so…

Booch: Were these people all across the world, or was it a smaller group? When did it start expanding?

Torvalds: It was never a close-knit group just because that wasn't how I had worked at it. Like, I had not worked with people at the university. So there were no face-to-face meetings. There never has been, really, from day one. We meet face-to-face sometimes over beers, but not for work. Right? I mean, the early feedback tracked computer use pretty closely, and Internet use, obviously. So it was all universities pretty much. All universities and a lot of-- even from the very beginning half of it was the U.S. And Finland did up getting fairly well represented not just for me, but because, I mean, Finland had fairly strong Internet use. But I think the East Coast in the U.S. was probably the best represented.

Booch: Any individuals or universities that come to mind or that are particularly passionate about this?

Torvalds: MIT seemed to. Well, it's hard to tell. Like one of the first people to send me a patch was from MIT. So maybe I just think MIT was unusually active for that reason. I haven't done any real statistics. But I think I kind of blamed it on the fact the West Coast had BSD. Right? The East Coast didn't have-- well, they had BSD but it wasn't something they had bought into as emotionally. So I think that may be one of the reasons why it was-- why I think it was MIT just because when you look at universities, looking at computer science and operating systems in the U.S. or in the world, it was like Berkley, and MIT and probably, I mean, other places too. But in Berkley they wouldn't bother with some small unknown start-up OS. They had their own.

Booch: Finnish guy. Which it just occurred to me, the original 10,000 line release, does that exist someplace?

Torvalds: It still exists as source code. We've lost a lot of the intermediate. So the first release I called 0.01 because it clearly was not 1.0 material. Right? I think we've lost 0.02, because I found a bug in that one so quickly that I released 0.03 like within two or three days. And back then 0.02 had no historical value to me, so I literally deleted it in order to not have people look at this horrible mistake I made, right, which is sad now. But at the time it seemed a very natural thing to do. And then our versioning system was always a bit flakey. I mixed numbers, with letters, with magical characters. I think I had a like 0.95C-plus at some point kind of thing. So we don't really have all of it, but we do have a lot of them.

Booch: And the we being…

Torvalds: Well, I just say we because it's obviously not been me personally who'd been involved. So it's the general kernel programming community.

Booch: We ought to make sure that it's kept in the [Computer History] Museum. One of the things the Museum has is a collections program for software. We're trying to collect the original source code to interesting things. So we have stuff like the original source code to MacPaint of all things. And I discovered that IBM's lost the source code to OS/360, so, oh well. When did you finally start moving over to the tens, the point something, the .1s, the .2s.

Torvalds: The .1 actually came fairly quickly because it turns out that 3.03, which I fixed the stupid bug, was a huge advance. I mean, the stupid bug had to really-- I mean, I was ashamed of it for a reason. Right?

Booch: What was the bug?

Torvalds: Back then the big deal was still the file system accesses. And the buffer cache had-- I forget what the exact detail was. But it had a, I think it was a reference count leak, which meant that after-- if you tried to compile something big and some file or block on the disc got looked at many times in succession, eventually the count would overflow, and the thing would be thrown away, and it would just use data and crash. And this is what meant that it could not actually compile itself under itself. So 0.03 was the first version that was self-hosting in that sense, that hey, you can actually compile-- maybe it was-- maybe it could compile itself, but it couldn't compile the compiler. Something like this. So C was a huge step conceptually forward, and it was so big that it actually ended up being quite useful compared to MINIX. I mean, not at the same level, but they were definitely approached. So at that point I said, "Okay, let's jump from 0.03 to 0.1." Actually I called it 0.10 just to confuse everybody. And that actually was the first release where people started to really-- I mean some people started to use it. I don't think anybody had actually ever used the previous ones. Maybe they had tried to boot them up, but they had not actually done anything with them.

Booch: Had anyone ported it beyond the ‘386 around that time or is that a little later?

Torvalds: No. No, and it was really-- I mean even on ‘386 you literally had to compile in the constants for your hard disk. You had to say how many cylinders and tracks your hard disk had. Well, I mean, it was one of those things where hey, I want to get it working, so I'll just hard code my own hardware on it.

Booch: Were you on Andrew's radar at this time with MINIX, or did they sort of view it as…

Torvalds: At 0.1, or 0.10 I think it started to be on that radar. Because now people are actually starting to use it. And part of the problem is, it was-- we didn't have our own mailing lists, we didn't have our own newsgroups. So any discussion about this was all on the MINIX lists, which was-- I mean that was a bit rude, let's face it. <laughs> So 0.10 was probably early in November or something like that. And eventually improved pretty dramatically. So I made a Christmas release, which might have been 0.12 where I actually had just added paging to disk. And that was a conceptually huge step because it used to just read the whole binary into memory with a read, and didn't actually use paging at all. So paging to disk was not just technically interesting, but it was also the first feature that MINIX really even had. So psychologically that was a big step. And I think I did paint in the swap space, too, so you could actually, even if you only had 2 MB of RAM, and some people had just 2 MB of RAM, you could actually compile stuff with Linux when you couldn't compile stuff with MINIX, at least not with GCC [Gnu Complier Collection], because GCC was a pig and needed more than two megs of RAM. These days, of course, it needs 150 megs of RAM or something, but that's a different thing.

Booch: The 0.12 release was also interesting in that you released it under the GPL license I understand. Tell me about that.

Torvalds: Well, the GPL, I think that actually happened-- I think 12 was released with the old license. But in the release, that was the timeframe where really people in the U.S. were starting to-- like at Unix users' groups they were starting to do disk duplication services where, I mean, you said, "Hey, I found this ploy on the internet." And people literally wanted to fit on three or four disks. You've the kernel. You've got some utilities there. And Unix users' groups would have people who did this for them. But the original license said, "You have to give all modifications back and you cannot charge anything, not even copying fees." Right? And that all harks back to the fact that okay, I was a poor student and I had thought that MINIX was way too expensive for me when it cost whatever, $160. So the source part was a big issue for me, but the price part was literally also an issue. So the first license literally said you can't even-- even if you, like, do disc duplication services you'd have to give out all that work for free. And people actually started emailing me and saying, "Hey, could I please recoup my expenses?" Right? I mean they didn't want to make money off this, but they wanted to at least not have to lose money on it. And when you get to that point and like, where do you draw the line. Right? Your time is an expense, too. You want to maybe get five bucks for the effort of having done this disc copying service. And I thought that was kind of-- that sounded perfectly reasonable even to me. And I realized that the whole cost issue was really not an issue at all, and it was unreasonable to really have that in the license at all. So I had been aware of the GPL and I had friends who were rabid about it. I was never really all that rabid about it, but it really fit my needs very well. And it was a well-known license. It was a well-known license especially of the one program I really kind of looked up to, GCC. Without GCC I couldn't have done any of this. So it was kind of also saying, okay, GCC is really important to me. GCC uses the GPL. I'll reciprocate, not by giving code backs in GCC, but at least kind of say that okay, you've used a really nice license, I'll do the same.

Booch: Had you trademarked the term Linux by this time? That came later?

Torvalds: No. No. I mean that is the kind of paperwork crap that I don't want anything whatsoever to do with. And the fact that I have the trademark now is just a huge pain in the ass.

Booch: Why is it a pain in the ass? People write to you for use of the trademark?

Torvalds: Well, no, we've actually set it up now, finally, so that you can basically self-certify most of the time. Well, there's these rules about when you can use Linux without any-- needing any permission. But it still comes up every once in a while. But it used to be a pain just because then eventually when Linux-- not that much later when Linux became more now, and I still lived in Finland, somebody else trademarked the name and started extorting people for money, basically. And then suddenly the fact that I had not trademarked it was-- I mean this would never have been an issue in Finland. I was still in Finland. Trademarking, why? That's not something non-commercial now. So people ended up actually taking this to court. And I had really nothing to do with it except for the fact that I was the only person who people could kind of agree to trust with the trademark, because all the people involved who really hated the trademark in the first place were all like distribution makers on a small scale at that point, and like the *Linux Journal*. And they all were kind of nervous about each other, but they weren't nervous about me, so I ended up getting the trademark just because of that. But it was just a horrible experience. And…

Booch: You mentioned the Christmas release in December '91, January '92. But then events really started unfolding then because you had the .95 version, the first Linux newsgroup and the first commercial distribution. How did that unfold?

Torvalds: Well, I actually don't even know how the commercial distributions unfolded. I mean they became legal with the change in license, and they were pretty small at the time. I forget which one may have been the first commercial. It might have been SLS which…

Booch: Do you remember who created that?

Torvalds: No.

Booch: That was the soft-lining Linux System?

Torvalds: Soft lining. I was wondering what was SLS? What did it even stand for? I couldn't remember. So I really didn't have anything to do with it. More importantly I didn't want to have anything to do with it. I was perfectly happy with people doing it. In fact I was really pleasantly surprised by the fact that, okay, apparently it's doing so well that people wanted to pay for the-- I mean, it was mostly floppy disc copying services at the time. And the only thing I cared about was that I wanted all the improvements to be-- to come back. They obviously did. Right? And the thing I cared about more was the fact-- and the thing that caused us to jump number again was not only did the license change after-- or with .12. but some people started putting X to Linux. And that wasn't something I hadn't really foreseen, because I'd actually not used X very much. We had xterm also at the university, but I'd mostly interacted with Unix actually over my regular serial line terminal. So I was like, the fact that we had graphics was something really unexpected to me, right, and not something I had ever really planned. So then people put an X to Linux, again mostly by actually building on the Linux capabilities. And that was such a huge step that I thought okay, we're getting close to a 1.0 release so I'll call it 0.95 instead of 0.13 which would have been, I think, the natural progression. And it turns out that I was a bit too optimistic.

Booch: In the sense that it wasn't quite ready for prime time?

Torvalds: Yes, in that sense. And we actually ended up…

END OF TAPE 2

Booch: Events in 92 where commercialization really began and I have in my notes Adam Richter was a gentlemen who did one of the first commercial releases.

Torvalds: Yes.

Booch: Tell me about him and how you encountered him and vice versa.

Torvalds: This is again, one of the fun things about Linux has been how I did not-- the whole Adam Richter thing and making a distribution, he just did it, and it ended up being called Yggdrasil. I don't remember if that was the original name, but there were like three or four different… Slackware, SLS, Yggdrasil.

Booch: Better spell that one because it's a curious-

Torvalds: Y-G-G-

Booch: D-R-A-S-I-L.

Torvalds: So it's the name of the Tree of Life in Norse mythology I think.

Booch: Whoa.

Torvalds: Which is why it's spelled so oddly, but I had nothing to do with that. It was not my choice. I think he chose a Norse name from Norse mythology because I was from Finland, which I mean if you're in the U.S. it's all the same. Never mind that Finnish mythology is completely different. So I think there was some forethought there, but kind of not. Off the track. But that's kind of been a common theme and one that I really enjoy how it has not been really planned. There has never been any central planning. I certainly haven't done any central planning, and a lot of these things just happened around Linux so the only interaction that I literally had with Adam was he would send me patches that he wanted, I mean he needed for his distribution. And that's actually one of the things that I found to be most important. All the commercial distributions, they had very different needs than the technical people, and it really added a whole new balance to the whole thing where if you only technical people who make decisions, the end result is usually not very good. If you only have marketing and non-technical people who do it, the end result if really horrible, but you need both. And that was one of the big advantages from the whole commercial approach was that it gave that kind of balance. I remember that I ended up rejecting a lot of the patches, because I thought that they should be done differently. But it still gave-- I mean quite often I ended up rewriting them and saying, okay, I see where you want to go, but I really don't want to do it that way. And I was very true early on that I kept a fairly strict control of the source code by just saying quite often instead of applying somebody else's patch, I would just say "Okay, that's a good idea, but I'm going to do it this way." And it actually took a while before I learned to just trust people. Partly it was trusting people in general, but partly it was trusting then specific people so much that okay, I said "This guy knows what he's doing. I'll just apply his patch and not worry about it." Obviously that's what I'm getting really good at these days and that's really my job.

Booch: Well, the level of complexity has grown so you can't go into those details.

Torvalds: Right, but it's also one of the things that I think some projects have a hard time sometimes accepting. Where a lot of Open Source projects tend to have a core group that really can't let go, and want to continue to control things at that level and it was something I did early on.

Booch: When did you start letting on? What year would that have felt like that you were letting go?

Torvalds: It's gone through many different phases. I mean it went to the point where I would trust some people so much that I would just say okay. When they send me a patch, I worked with them now long enough that I would say okay, this guy, he knows how I work. I know how he works. I'll just apply it. And that's kind of the first stage, but then especially as we started upgrading on our tools and I ended up at some point being really the limiting factor in applying patches from people. We ended up having this much more distributed system where now I just don't apply patches from people I trust. I trust other people to apply patches from people they trust so it's this network of trust, and when they say "Hey, take this whole set of patches," and you say "Hey, I trust you." And it's been an evolution. In fact it's been one of the biggest things that's been evolving is just how you work together with people across the globe, while still, you have your balance. You have to take a new code, because if you don't take a new code, people just get really frustrated and they just say there's no point in me working at this, but at the same time you do want to have-- I mean you need to have quality requirements, and balancing those two well is really, that's probably the biggest problem we've had, and I think we were doing really well on it considering how things- development is going. But that's been one of the important issues.

Booch: So back in the earliest days what machines were you now developing on, because you probably moved on beyond that first piece.

Torvalds: It was still PCs but by then I mean we started expanding on support for different kinds of hardware. Not just you didn't have the right [ph?] code or the size of the hard disc anymore, and you could use SCSI hard discs instead of the ST506 controllers and IDE and things like that. And the CPUs were obviously also getting support for more modern CPUs from Intel.

Booch: How were you supporting yourself during this time frame?

Torvalds: I was at University. In Finland, University is free. Well, you have to pay for extended health care which is fifty bucks a year.

Booch: Or a few copies of Linux, how nice.

Torvalds: Yeah. And by the way, you also get these basically now, you get money from the state to support you, and you get a basically no-fee loan too. Finland has a really good educational system. If you want to be educated you can do so. You don't have to have money. You don't have to have an income.

Booch: And you were teaching a little bit too.

Torvalds: I ended up teaching also first just being like a TA and eventually actually doing some introductory courses but early on I was just a plain student.

Booch: Which is how, by the way, you met your wife I understand in one of the courses you taught. Can you relay that story?

Torvalds: You're not supposed to do that, I guess, but I married her so I made an honest woman out of her. Where would I meet people otherwise in my basement on my computer? There were no dating sites back then. So yeah, I met my wife at University.

Booch: There was also in this time frame that you and Andrew got into a bit of a tiff I understand over microkernels versus-

Torvalds: That was fairly early, yeah. For some reason people think-- I have this name of being a nice person because people, I don't know. But at the same time since the beginning I've had very strong opinions, and I've always enjoyed arguing with people, so my way of writing emails and my way of expressing myself is actually not all that pleasant. I am argumentative, and I don't like being proven wrong, but I'm pretty happy to say something pretty extreme and then somebody points something out and I said "Okay, you're right. I was wrong. It's not quite that extreme." So we had a lot of flame-wars [ph?] over the years. We still have a lot of flame-wars. The most famous one is probably the first one when I told basically reacted badly to Andrew Tanenbaum telling me that [what] I did was worthless and bad.

Booch: He had strong opinions too.

Torvalds: Yeah, he had strong opinions too. I just replied not very politely saying that "He shouldn't complain about other people's code when his own code is so horrible."

Booch: He did say some provocative things.

Torvalds: I answered in my not-so-polite way. Sometimes it's me who starts it so I'm not going to-- in fact we've met since and we've had email correspondence since. I actually still think his book is one of the best books about introductions to operating systems. It's very down-to-earth and actually having examples is a great way to teach people in general, and a lot of other operating system books are not. They're talking about concepts and not explaining them very well.

Booch: Perhaps it would be fair to say you were both very passionate people.

Torvalds: We are. Yeah.

Booch: You had a similar run in with Richard Stallman as I understand over the whole <inaudible> things.

Torvalds: Also another very passionate person.

Booch: Yes, Richard is indeed.

Torvalds: Yes. You know, we've never really gotten along, although we've usually been able to not butt heads too strongly either. So with Richard the problem has always been that I've been a plodding engineer in Richard's mind, and Richard to me has always been this religious fanatic. We've clashed over that a few times.

Booch: You've clashed over naming as well too, I understand.

Torvalds: I tried to let that thing go, and he wanted everybody to say "GNU Linux" instead of just "Linux." I said "Whatever. You can call it whatever you want. I'm not going to go there." And I let that whole thing go. I don't think I clashed too much over that. We've clashed over other things. Engineers or software programmers in general seem to be pretty headstrong, and then when it comes to things like licensing, people get really excited about that.

Booch: And then as I look at the next few years there was this growth pushed by commercialization because it really took off over the coming years. You had Debian, you have Red Hat… come on to the scene. You still however were still pretty much in control of the Kernel around that time.

Torvalds: Yeah, I mean part of it was again that everybody was perfectly happy having somebody to integrate it all. And a lot of the work that the distributions ended up doing and all the commercial efforts ended up doing-- I mean much of it was in the Kernel, but at the same time a lot of it wasn't. A lot of it was in making installation easy. A lot of it was in the programs around the Kernel. And sometimes those programs needed Kernel facilities so that obviously there was this feedback cycle. But there was never really any conflict there. And part of it was that I very consciously did not want to work for a Linux company. I didn't want to make it so that let's say I'd work for Red Hat and then that would-- even if I would have stayed neutral personally, it would still have given this appearance of non-neutrality.

Booch: You were certainly asked, I imagine, but you resisted their advances?

Torvalds: I think everybody was so happy with the situation that even the distributions don't-- I didn't think they really even wanted me to work for them because they see the advantages too. And I was actually-- I got this offer from Transmeta [Corporation] which suited me perfectly because the whole reason I got into doing Linux in the first place was that I was interested in low-level CPU details and here comes a CPU company and says hey, "Do you want to work for us and by the way, we're even lower level than anybody else. The things you need to do are kind of below the surface."

Booch: What year would Transmeta have been?

Torvalds: I started in '97 and Transmeta was founded I think in '95, but when I joined it was really pretty small.

Booch: And they pretty much gave you license to just go do Linux, basically?

Torvalds: It wasn't even that. Part of my contract said that Linux is not my job. I'm allowed to do it but when I do it, even if I do it on work time, it's still incorporated [ph?] to me. And that was just again-- it had already grown big enough, and it was important enough to me personally that I wanted to make it clear that this was my project and I could do Linux on Transmeta time because people were already back then using Linux a lot in the whole EDA business, so doing all-- when you're a chip company, these days Linux is one of the major platforms for doing design, but it was already becoming fairly popular. If you're a highly technical person and you wanted a Unix workstation for free or cheap, you'd go with Linux.

Booch: In fact, in the '90s when did you discover or have this revelation that there was this crossover that all of a sudden people [were] using Linux for real things? Did that ever hit you in that timeframe?

Torvalds: Actually, it wasn't ever a big deal. I mean I think when Oracle announced that they were porting Oracle to Linux, I forget when that was, but that was like around '99 or something like this. Maybe it was earlier. That was a big deal because that was like Oracle was kind of the definition of a Unix vendor. If you didn't have Oracle, you weren't a real Unix vendor. So that was kind of a big event that I remember reacting to, but at the same time, most of this happened so gradually. And a lot of it was like even when it was a big thing you could kind of see things leading up to it, so the only point I was really surprised at, it was literally like late '91 and early '92 when it went from my personal projects to hey, there's actually 10 people using it or 100 people using! That was a much bigger step. Then everything else was not that big of a deal, because it wasn't what I was interested in anyway.

Booch: For Transmeta, that's when you relocated to the U.S., to the Bay Area?

Torvalds: Yes.

Booch: Was that a bit of a shock to come to the Bay area?

Torvalds: It was and it wasn't. It was- we did it I don't know, it probably wasn't the smart way. But my first daughter was born ten weeks before we moved, so I was a new dad, moving to a new country to my first real job. I mean I had had a job at the University but let's face it, being a-- that was still a fairly gradual thing and this was like my first job out of school; new language, new culture, having to buy a car, realizing later on that you have to buy two cars. So it was a big thing but I really think that on the whole because of Patricia's birth, that was a fairly chaotic time anyway, so in retrospect it probably worked out pretty well and apart from remembering having headaches for the first few weeks, it is all a blur. But it was very good. It was a really nice decision to make, and what we reacted to later was we met some people who had also moved from Finland to the Bay Area and they were working for Nokia who had opened an office there in <inaudible> later on. I reacted; my wife too, reacted very strongly to how they moved completely differently. They knew that they were moving for two years or three years, and then they would move back. We had this whole thing, "We'll move to the U.S. and see how it is." And I think we integrated a lot better because of that, because we didn't have this knowledge of we'll move back. It was more like okay, if things don't work out we'll move back, but we didn't have a plan, so it was-- and Finland is not that culturally different. Apart from the whole car thing and the language thing it was not a big shock.

Booch: When did Linux get the penguin mascot? When did that sort of hit the scene?

Torvalds: We did it for one of the releases. I forget which one. I don't think it was 1.0. It might have been 1.2 or 2.0. When people-- it was getting fairly popular commercially and people really felt that we needed a logo.

Booch: Did you think it was a crazy idea to need a logo?

Torvalds: No, I thought it was a good idea. I thought it made sense to have a logo and unify things around something. But some of the ideas were pretty crazy.

Booch: Like?

Torvalds: A number of people, for example, felt that okay, since we have this wild and crazy name of Open Source and being university students and things like that, the logo should be really corporate and staid to kind of balance it out, so some of the logo suggestions were really pretty boring. And I had kind of the opposite reaction where I wanted to have something cuddly, something friendly, something non-scary.

When I actually visited Japan later on I noticed they had started using a logo with a very cuddly octopus, which was a wonderful logo because it fit my kind of requirements really well, and apparently for some reason octopi are considered to be friendly in Japan. Whatever.

Booch: I just eat them.

Torvalds: Maybe because they're good eating, I don't know. So they ended up using a different logo than the penguin, but I wanted something, an animal. If you want something cuddly, it needs to be like an animal, but you can't have a dog as a logo because I mean that's not very interesting, right?

Booch: I remember reading an email though of how you characterized what the penguin should look like.

Torvalds: I was basically at some point I decided okay, it should be a penguin and that was probably influenced by there's this movie studio. Well, it was not a movie studio at that point, Aardman Studios.

Booch: Oh they do the claymation stuff.

Torvalds: They do claymation stuff. And they had-

Booch: *Wallace and Gromit*.

Torvalds: They did *Wallace and Gromit* and now in the U.S. they're better known for <inaudible> is *Wallace and Gromit*.

Booch: Yes that was a good one.

Torvalds: But before that there was the movie about chickens, *Chicken Run*.

Booch: Oh *Chicken Run*. I can tell you have daughters.

Torvalds: But even before that they used to do these commercials for British TV using clay animation, and one of them had this fat penguin, and I saw that. Somebody sent me a picture of that and I saw that and said "Okay, that's more like it. That's what we want to have." I didn't want to use somebody else's penguin, obviously, and I can't draw. My artistic talents are a perfect zero. So I basically said "What we want is this cartoonish penguin, sitting down, fat, gorged on herring, and just very happy and cuddly." And that's how somebody who could draw actually made the logo. I had like hundreds of people send me logos, most of them pretty horrible. But I'm really happy with it. Not everybody was. People really felt that it was not very professional. Let's face it. It's more like it's not really even a penguin because it actually has an orange beak and orange flippers and penguins don't have that. It's more like a duck that was raped by a penguin or something, right? It's this cross-species thing.

Booch: You gave it the name Tux?

Torvalds: No, I didn't. Somebody else gave it the name Tux. It didn't have a name as far as I was concerned. The name just stuck. It was a penguin. Calling it Tux makes perfect sense. I'm not going to argue against that.

Booch: So you were with Transmeta from when until when?

Torvalds: I was with them from '97 to '04 or something like that, which was unusually long in the Bay Area. People seemed to change jobs about twice a year or something. Well, maybe the downturn kind of put a damper on that, but it certainly felt that way. But I've always been the kind of person who I'll do something and I'll stick with it. By the end I was one of the senior people there just because most people didn't stick with one thing in the Bay Area.

Booch: You mentioned Oracle and their announcement to port over to Linux and IBM did something interesting too around that time. Tell me about that.

Torvalds: The IBM announcement.

Booch: What was that announcement?

Torvalds: IBM had been fairly active in Open Source for a while, but then at some point, I forget exactly when this happened, they announced that during the next one or two years they were going to put a billion dollars into Linux or something like this. I forget the exact number. The details escape me, but it was like a big number.

Booch: Did that surprise you?

Torvalds: That did, yeah, I mean I think it took a lot of people by surprise, and I have to admit I was pretty skeptical. I mean when it comes to companies making PR announcements it's a lot easier to do the talk than walk the walk, right?

Booch: Very true.

Torvalds: And so I was okay, let's see how that works out. It did not seem to be a match made in heaven, let's be quite honest. My personal view of IBM was this pretty staid company. I think most people would agree that was kind of an odd bedfellow thing. But I have to say, I mean, that they really followed up on it, and they put a lot of engineers into it, and they've been, I mean I was surprised, and I was wrong. I'll freely admit the fact that that really did work out really well.

Booch: Around this time too Linux was starting to make inroads in the embedded space. Did that surprise you? That was not its original bent.

Torvalds: It was not, but I mean a lot of the things that-- the embedded space was funny. It really, some of the really early things surprised me because I remember it was one of the first *Linux World*s or something that some embedded people came up to me and told me how they had worked on this gas pump that was running Linux because it had a screen on it with like web browser or a TV or whatever. I forget the exact details. And now you can actually see those things, right? But this is probably late ‘90s something, and the whole notion of running Linux on a gas pump, I was like okay, your guys are crazy, but in a good way, right? And there was a fridge also.

Booch: A refrigerator running Linux?

Torvalds: A refrigerator with yeah again LCD screens so that when you’re really bored instead of just opening the door you can just stand there. Maybe in between, this is the U.S. In between opening the door and snacking you can just stand there and drool in front of the fridge, so some of that took me by surprise. At the same time a lot of the other embedded spaces where the actual use wasn’t quite as lackey [ph?]. I mean, Linux actually made tons of sense because the whole not having to license fees and things like that, and the flexibility. In that sense the embedded space and Linux make a lot of sense.

Booch: So, in the beginning you mentioned the issues of the influences of the technical and the commercial kind of really impelling in certain ways. Did the move of the embedded space also push you in certain ways with the Linux release?

Torvalds: To some degree the embedded space has been very problematic. I think that’s cultural. Embedded people are-- they’ve had-- it’s been getting a lot better over the last couple of years but especially early on, so what you would have is the embedded manufacturers, they always, I mean they have some specific platform in mind. And then they port Linux to it. But they are, to them, Linux is a necessary evil that they need in order to get their real work done, so they would usually do the fastest, ugliest port that they could possibly do, and not in a way where I mean they make their changes available, but they wouldn’t really be useful to anybody else. And then two or three years later, they’d do it all over again, and start basically from scratch because two years later the Kernel they wanted to start with was completely different from what they had done last time around. So the embedded space-- and I think that’s what they were kind of used to doing. The embedded space has not generally been all that great at buying into this whole incremental development and release often, release early kind of model. Plus the hardware and the environment tend to be so specialized that it’s not that relevant to anybody else. That’s changing partly because the embedded space is growing up.

Booch: “Growing up” in what sense?

Torvalds: In a pure hardware sense. It’s too expensive to make. In the embedded space, it used to be that you did everything custom. It’s just too expensive to do that. In the embedded space today it’s much more reasonable to just take a general purpose CPU. They used to be too expensive or too power hungry or something, but hardware has advanced enough that most of the time when you get an embedded board, it’s going to be a big standard ARM core, and there’s going to be some specialized stuff around it, but basically the embedded space has come to where the PC was ten years ago.

Booch: If I’m not mistaken isn’t the Mars Pathfinder running Linux?

Torvalds: It may well be so. One of them was running BXware [ph?].

Booch: I’ll have to dig that one up.

Torvalds: Linux has been in space on several missions. Whether it's on the Mars Pathfinder or not, I'm not sure.

Booch: What was the first thing in space you recall?

Torvalds: It was just somebody who took Linux up on the Shuttle for some experiment or other.

Booch: Sweet.

Torvalds: But it was a big deal at the time because again that was some time ago. It was like okay. I was surprised people were using Linux at all. It was early on.

Booch: So almost at the end of professional things, and then I have some general ones. You're now here in Oregon because you're with the Open Source Development Lab.

Torvalds: No.

Booch: Oh you're not.

Torvalds: Well, I am here in Oregon, but the causation was not correct. I actually started working for OSDL [Open Source Development Labs] back in California and part of the requirement was I don't have to move. I was planning on taking a hiatus from Transmeta because I wanted to concentrate 100 percent on Kernel release 2.6 [which] was coming out. And Transmeta was actually, they were really good in every way, and they said you can just take a break and do it as a Transmeta employee and we won't mind. For a while at Transmeta I actually was basically doing 100 percent Linux, but it just made me feel guilty so I said "I don't want to work this way anymore. I feel stupid working for Transmeta and not actually doing the work that I was hired for originally just because I needed to concentrate on Linux. So I said "I'll take a break. I'll take an unpaid leave of absence for a year." And OSDL came in and said "We'd like to sponsor your unpaid leave of absence for a year," and I said "That's good." The whole health care situation in the U.S. means yeah, I'd rather be employed than unemployed, let's put it that way, even if I could afford the year of absence. Otherwise, but basically what I wanted to go back to was basically where I started Linux where I felt like I could do whatever I wanted to and I didn't have to feel guilty about that. I hadn't wanted to work for a Linux company, per se, because I didn't want the polarization, but working for Linux Foundation, or back then it was OSDL, it didn't feel like I was making any other commercial companies feel any worse, so it was kind of a very good suggestion. So I was working from home for a year for OSDL and then we moved up here because we started realizing that our kids didn't actually know anybody in their neighborhood, and I don't think Silicon Valley is very nice from a social standpoint. A lot of people move there for a couple of years. Almost everybody who lives in the Bay Area, you talk to people and 90 percent of the time they're not actually local. Well, they may live there but they didn't grow up there, and so we started looking. I could work anywhere.

Booch: <inaudible>

Torvalds: We actually looked in Southern California because I-- the weather. I like scuba diving. Maybe I could work there.

Booch: You're a physical guy. You scuba dive.

Torvalds: Yeah, I do that once a year, because you can't scuba dive in Oregon. Well, you can but it's not very good.

Booch: Little cold.

Torvalds: I don't mind the dry suit. It's just [that] there's nothing to scuba dive in.

Booch: So a bit of a divergence, where do you like to scuba dive? Where are some of your favorite places?

Torvalds: Usually we just go to Hawaii not because the diving is better than other places but because it's easy to go there with kids especially.

Booch: That makes sense. There are some good spots.

Torvalds: There's a direct flight both from the Bay Area and from here.

Booch: Some good spots in Maui to dive.

Torvalds: Maui is where we go. I have no idea how we even got here.

Booch: A bit of a divergence. You mentioned scuba diving.

Torvalds: Anyway so we were looking in Southern California, but I don't know why. Probably social reasons really we ended up here, and part of it is probably because it turns out Portland is pretty much exactly the same size as Helsinki, so maybe I'm just more comfortable with that kind of size of city.

Booch: And the weather is equally gloomy?

Torvalds: It's different. The weather here is better than Helsinki. Helsinki gets a lot colder. On the other hand, then during the winter snow actually raises the albedo on the ground so it's actually some degree a bit brighter. It rains more in Helsinki because it rains-

Booch: It rains more in Helsinki than Portland! Wow!

Torvalds: Yeah. Well, Portland has more gray and drizzle for six months of the year, but so it doesn't rain really that much.

Booch: So what's a typical day in the life of Linus these days now working with the group?

Torvalds: The typical day is I get up, during the summer I'm not a morning person, so during the summer when the kids don't have to go to school it's usually around 9:00. Get up and go downstairs in my bathrobe. I don't even get dressed. I commute all the way by walking downstairs here, and then I usually sit there until noon or so, at which point I get showered and get dressed and take a short break and then I go back and sit read email some more. That's really what I do. Literally the only thing I do is I read email. I integrate the work from other people.

Booch: Do you do much programming now yourself?

Torvalds: I don't. I do some but that's not what I consider to be my main job anymore. For a while, like a year and a half ago, I did a fair amount of programming, a year and a half, two years ago because I ended up doing my own source control management system, and I actually did programming in user space just to maintain the Kernel sources, so I'm happy to say that CVS [Concurrent Versioning System] is actually going away and my system will replace it.

Booch: Yay, and that system is called?

Torvalds: Git. The joke is I name all my projects after myself. First Linux then Git and all the Americans say 'what's the joke?' But 'Git' is British slang for a despicable person or unpleasant person. It was really just a three-letter combination that's easy to type.

Booch: So what's your current machine? I'm sure you're running Vista, right?

Torvalds: Right. My current machine as of the last few days is … one of the nice things about doing operating systems is all these system vendors want to give you like early heads-up. So right now I'm running this unreleased <inaudible> setup from Intel that it's a sweet machine. You get the new hardware to make sure it works and that there aren't any huge surprises, and quite frankly I mean most of the-- by the time I'm ready to take it, other people have already made sure that it basically works. I used to enjoy doing bring-up,,but I don't have the time anymore, quite frankly.

Booch: To what would you attribute Linux's amazing success? How did it explode?

Torvalds: It's all me. We need to change the tape?

Booch: In a minute. We can change it now? We can change it now. Let's change it now.

END OF TAPE 3

Booch: So, the question was in retrospect, why do you think Linux has been so wildly popular as it is?

Torvalds: I'd like to say it's all me. I mean, a lot of it was just being in the right place at the right time, doing the right thing. And part of it is, so, like, traditionally Unix, even the free Unixs that did exist and, I mean, still do exist I think their mindset was, you know, very traditionally Unix gutter really. They were doing the whole-- they'd given up on the desktop. And to me the desktop was the only thing that was interesting because that's what I wanted to have. And Unix was in this server rut than just-- including the free ones, I think. And so Linux had the same good, like, technical foundations in a concepts kind of way. But at the same time had a very different way of doing things, not just development model, but also the kind of people we reached out to when the free Unixs tended to reach out to all Unix hackers. Most Linux programmers tend to come from a younger crowd, but also more of a DOS/Windows kind of background that we're just looking-- I mean, they usually had some including about Unix. But it was more of a-- I think it was a culture that was slightly more accepting of DOS and Windows actually. Even though, then, Linux is known as this anti-Windows crowd, I don't actually think it's all that true.

Booch: What's your opinion on Windows, speaking of being opinionated?

Torvalds: My opinion is I don't actually care because I've never really had to use it. I never, I mean, it's…

Booch: What do you think it'll take for Linux to really break through on the desktop to get really encroaching upon the market share of the Windows role?

Torvalds: One thing is just lots of time, I mean, which I was a bit naive about too, clearly. But one of the issues is just-- especially the desktop, there's just this huge mindshare for people to change. And it's quite easy to just go with the flow and all the networking effects definitely favor the incumbent. And Microsoft has really needed to screw up as badly as they have lately and Linux just needs time to get more mindshare. I mean, there's a lot of technical issues too like all the programs and things. It turns out the Web helped a lot there because a lot of the programs-- one of the strengths of DOS and Windows has been all these odd programs. I mean, there's millions of specialized programs for DOS and Windows and they're to some degree going away. Because a lot of these specialized programs, like, people used to maintain their-- I mean, tax preparation was a big deal right in the U.S., balancing your checkbook, all these things, and there's millions of them. But it turns out a lot of them are done over the Web now anyway. So, that kind of made it easier for people to get into the whole desktop space. And-- so, that's one of the things that has been helping out a lot.

Booch: And the whole abStack. I mean, Linux is pervasive. You speak of the Internet, Linux is pervasive in that space.

Torvalds: You know. But on the other hand, that abStack tends to be more of a traditional Unix abStack. So, a lot of the desktop applications have actually been written for Linux or with-- not necessarily exclusively for Linux, but have Linux as being a big part of that, including very much like the windowing-- the Window manager environments that people use today. And, let's face it, Unix used to have horrible designed-by-committee Window managers. And Linux really ended up being the thing that revitalized that whole space. Before Linux there was-- people were giving up.

Booch: So, you were a little self-deprecating at the beginning, but I think you have to admit that Linux wouldn't have existed without you. Your managerial style and your leadership style are utterly unique.

Torvalds: There's a few things that I think do work and one of them is not caring too much, oddly enough. It's like I think what does not work and what a lot of projects end up doing is that they care very much about one specific model, right. You have an overarching goal and you want to push that one thing and Linux never had that. In fact, I mean, from the very first day it was like, okay guys, can you please give me something to work on because otherwise I'll be bored and I'll be done with my project. I want people to give me feedback. And I think that was really important because a lot of people start doing-- they have some design decision that they're attached to and they do everything according to that design decision. And it turns out reality doesn't work that way. Reality is not black and white. And if you actually want to be useful, you need to be able to just say, okay, that works most of the time. But we'll need to do this ugly other case for people who don't want to work that way.

Booch: Would it be fair to say that Linux' architecture is more accidental than intentional?

Torvalds: I wouldn't say accidental. I'm a huge believer in biology and evolution, right. And I would say organic…

Booch: Oh, organic.

Torvalds: …in the sense that the one driving goal has never been this vision. It has more been, okay, what do people need and what works, what doesn't work, and being flexible in that way.

Booch: So, in the history of from the point zero aught versions to where you are today, can you identify major architectural transformations or was it really an organic evolutionary process?

Torvalds: Yeah, there hasn't really been any. I mean, part of that has been that obviously we started with a very solid pace. Unix itself had a few guiding goals like the whole-- everything is a process and everything is a stream of bytes. But at the end of the day when you look at what actually happened over the history of Unix, yeah, everything is a process and everything is a stream of bytes except when it isn't. So, Unix itself, over time, kind of got more of the dose of reality. And then Plan 9 tried to re-imagine the original beauty of Unix.

Booch: Plan 9, to get this on tape, that was to … tell us the individuals involved on that one?

Torvalds: It's all the Bell Labs people and who's the-- I mean, Ritchie was involved.

Booch: Was it Ritchie? Dennis Ritchie, yeah.

Torvalds: But I think Dennis Ritchie was actually more of a manager of the team and-- <child in background> what's he saying? I met them. They're incredibly bright people, but I think they were the example of people who are a bit too-- they have too much of an agenda, where being clean is very important and sometimes more so than be practical. So, that is one thing I've, I mean, I've tried to be very pragmatic in everything, including…

Booch: So, that's more of a guiding principle for you?

Torvalds: Yeah. The whole pragmatism. And also including when it comes to the whole license thing I think you don't have to be so black and white when it comes to licenses. I used to use this commercial source control management program for maintaining the kernel just because it was the best one out there. And it worked wonderfully well for three years until things didn't work anymore.

Booch: Which one was that? What commercial program?

Torvalds: It was a program called BitKeeper.

Booch: And now you're moving to Git.

Torvalds: And the reason I wrote Git was that there ended up being all these horrible political and frightening-- and then license discussions and it was just-- at one point it was like, okay, I'm walking away from this. And quite frankly, everybody in the whole source control management system world is a complete moron. And I can do something better in two weeks and I did. And three years later somebody else is actually maintaining what I did. But it was one of those things where part of the problem had been that I was very flexible. I didn't think that licensing was something to be religious about. Let's do the best thing we can do. And I chose a commercial program that was free to be used in the sense that it didn't cost anything for open source programming. But…

Booch: In addition to pragmatic as a guiding principle, it also sounds like from what you said that a drive towards simplicity is very important for you as well too.

Torvalds: It is, but within reason. I mean, you can certainly take simplicity as one of those guiding goals and just taking it way too far to the point where it's so simple that it's not useful anymore so. So, the other thing I've been good at, because I don't care is the letting-go part that I already kind of alluded to, where it's too easy to try to be controlling. I'm really lazy. I consider it to be the-- like the primary goal in my life is literally to sit on the beach sipping a margarita, going out for a scuba dive. I mean, that's what I want to do. I'd be bored to death doing that, but it's kind of my mental model of what everybody should strive for. And part of that is, hey, if you can actually make things easier for you, go for it. So, all my workflow has been so automated I can merge easily tens of thousands of patches in a day. And in order to do that I need to trust people and I need to be able to find people who are worthy of being trusted. But…

Booch: In addition to that release process, what's been the testing philosophy for Linux as well, too?

Torvalds: It's always been okay. If it compiles, ship it and let the users-- I mean, I say that. It's actually true and not true at the same time. Because it's this multiphase thing where we actually have different levels of confidence in our trees. So, individual developers have their own thing that they're working on and they should release it openly and talk about it openly and try to get people to use that. But on the other hand, they should not try to push it to me until it has been tested by others. So, that's kind of like the CERUF [ph?] level of testing is even before it comes even close to me, it should get as wide a testing as possible. But then when it comes to me, we have a separate merge window when we say, okay, this is when we do merging of new features. And that's two weeks out of our release window that is two-and-a-half months. So, really, just 20 percent of the whole release window is for merging new stuff and 80 percent is for testing and finding regressions. And if we find a regression that we can't easily fix, the rule is we remove that feature immediately.

Booch: Are you driving the release schedule?

Torvalds: Yes. And it's actually-- well, I'm driving it in the sense that I'm the one who says, this is when we do the first release candidate. And then I decide usually two months afterwards, okay, this is released, but it's all based on time. We don't…

Booch: What release are we on currently?

Torvalds: Right now we're on 2.6.26 and we're working towards 2.6.27. And I'm actually in the end part of the current to merge window so.

Booch: How often are you scheduling releases? What feels right?

Torvalds: It's not what feels right, it's what works. And what we found is what works is two weeks of merge window because one week is too short of a time. Because that means that everybody has to synchronize and everybody has to have everything lined up just right and nobody can really conflict. So, one week isn't enough to merge, but more than two weeks would be too much because then people have time to start trying to do a new development and they're not supposed to do that. They're supposed to just merge what they have already worked on. And then two months of stabilization because longer than two months people start losing interest. And shorter doesn't work, so.

Booch: Sure. What's going to trip you over to the 3.0 version and then a 4.0 version?

Torvalds: The way we're working now, nothing. Except maybe that we just decide, okay, we'll need to update the major number just to make the minor numbers be smaller, right? That's literally how it's working. We have stopped doing releases based on features. We do releases based on time and the features that come in are the features that are ready. It just doesn't work to have a plan saying, okay, the next release needs to have these five things because that is what people are asking for. It doesn't work because it turns out that out of the five things, only three are ready. One turned out to be a really bad idea to begin with, and one might make it to the next release, but probably won't, kind of thing. So, if you do a release based on features, you're stuck and you end up with a mess that Microsoft is in because they apparently do that. And then you have release cycles that are three, four years long. We used to do that and we stopped just because it doesn't work.

Booch: What's a typical release cycle from beginning to end?

Torvalds: The release cycle is literally two weeks of merging and then…

Booch: But, I mean, how many times a year would you do it?

Torvalds: Oh, we'd have it three, four times a year. So, the total cycle is two-and-a-half months, basically. I aim for two months. Two-and-a-half months is what I consider to be realistic and then bad releases when we can't fix it up in time, take over three months. So, that's why we don't actually get four releases a year.

Booch: So just a handful of a few more topics. What is Linux going to look like five years, ten years from now?

Torvalds: I don't even think that way.

Booch: Really?

Torvalds: No, no because most of the things that push us are external. There are very few things that, okay, we have this thing that we want to push on people. It works the other way around where the whole point of an OS, as far as I'm concerned, is really to be the layer between the hardware and the real software. The OS doesn't do anything on its own. That's my personal opinion. That's how it should be. And if your OS does something on its own, you're doing something wrong. And that means that the OS depends mostly on how hardware develops, which clearly happens all the time, right? So, that's not going to end any time soon. But also very much how people use it and what capabilities the programs need from the hardware that the OS just has to kind of juggle. Sometimes it's not hardware, sometimes it's just prediction domains between two pieces of software. But that just means that we don't look for features we want to implement. We look for features we can do better that we already have or more commonly, we just have input from hardware or from users saying, hey, we have so and so problem and I can't do this. And that's what tends to drive us.

Booch: Are multi-core chips sort of pushing you in a certain way right now?

Torvalds: Well, right now I don't think it's realistic to-- I mean, it's-- we pushed our scalability up to-- I mean, people have been selling Linux machines with 512 CPUs for a long time and in some cases more. We're pushing that up, but it's-- so we have done patches to merge it so that it works better on… if we 4,000 CPUs and we'll continue to push that up to 8,000 CPUs. But realistically, that's still going to be the special case.

Booch: Multi-core is a special case?

Torvalds: Well, not multi-core per say, but going into the thousands. Most people would have four or eight and that's kind of-- we've shifted what we optimize for over time. It used to be we optimized for a single CPU core. And, okay, so that's not realistic even in the embedded world we have-- two is pretty much the norm now. In a couple of years four will be the norm. My new machine has eight. So, it's not a huge issue and I don't think it's realistic to-- software-- doing threaded software is too hard.

Booch: What makes it so hard?

Torvalds: It's not how people think, or actually it turns out it is how people think in the sense of that's how the brain works, but it's not actually how you think as a programmer. Very few programmers can really think in a parallel world. It's really hard to do.

Booch: Do you think this is a limitation of the human brain or it's an issue [of] us needing to do better languages?

Torvalds: I think it's a limitation of the human brain because it's a limitation of our language in the sense of our human language. We are linear thinkers because we think in languages. We don't think in some parallel model, right? At least our conscious thinking is linearized by our linear language view, which means that fundamentally the only thing where you get parallelism is if the problem is really parallel or if you find a way to turn human linear thinking into parallel thinking. And people have been trying to do that for the last 30 years and have generally been very unsuccessful at it. So, it's true that certain problems and, I mean, video and graphics in general tend to be extremely parallel and so trivially at parallel that it's easy to think about the parallelism. But I'm not a huge believer in the multi-core as a general purpose of view kind of thing. So, we'll see what happens. I've been wrong before, but I don't think I am when it comes to this.

Booch: What are two of the things you've been wrong about?

Torvalds: Okay, the really embarrassing one and which is kind of related to this one was I did not think SMP was going to be realistic on the desktop at all ten years ago. And the reason was is that it was also expensive. And then Intel turned the tables and just integrated everything on their die and suddenly doing two CPUs was almost as cheap as doing just one. So, hardware advances do change and I-- who knows, I may be wrong about people being able to take advantage of many, many cores.

Booch: So, what are two things you've been really, really right about?

Torvalds: I don't know. There's a few things I've been really happy about technically. But on the other hand, there haven't been any things I've been really unhappy about either because we can fix the technical side. So, maybe the thing I've been really, really right about was really the whole license choice. And so I think that has been the thing that really mattered most.

Booch: What makes Linus happy? Where are your passions?

Torvalds: I don't know. I've been really happy about the fact that my job is my hobby, right, or my hobby is my job. However you want to put it, right. And the fact that what I do is not just fun to do, but it also seems to be meaningful, so it actually matters to other people. And if you have those two things, I think you'll be happy whatever you do. I mean, it doesn't have to be computers, but within computers I've-- that's what I ask for. That's what I hope my kids will have, although I don't know what they…

Booch: Do you think they'll be programmers?

Torvalds: I don't know. It doesn't really seem to be that way, but I don't care. It's not like I'm going to push them that way either. I just hope that they find something that they are really interested in.

Booch: In fact, what advice would you give to the high school or college student who wants to follow their passion? Do you have any words of wisdom for them?

Torvalds: Well, if you want to follow your passion, do it. Right? But on the other hand, I mean, a lot of-- that's a dangerous thing to say because a lot of people's passions are really stupid. I mean, let's face it. It works when your passion makes sense, but quite often your passion isn't really going to be your work anyway and there's no way. I mean, you may be passionate about any number of things that just aren't worthwhile.

Booch: Won't work out.

Torvalds: Yeah. I mean, so, it's kind of-- it worked for me, but I'm not going to say it's going to work for anybody else.

Booch: So, you actually may have already answered this, but if you hadn't sort of discovered software, what would you have been doing do you imagine or can you not imagine a life without this?

Torvalds: I don't know. I have been doing programming for so long that it's kind of hard to imagine what else I would do. I mean, I know what I'm interested in and I would probably-- if it hadn't been computer science, my guess would be I'd be doing biology. And especially something like I'm to do with DNA and evolution, and like biology on a low level, the same way I enjoy doing computers at a low level. I would not be the kind of person who actually deals with the animals. I would be the kind of person who deals with cells and then things like that, I think.

Booch: You have to be proud to realize Linux probably runs most of the DNA sequencing machines these days.

Torvalds: Yeah. Well, except the ones that are special built and don't have any OS at all because they're just doing the calculations, yeah. But yeah that's…

Booch: So, in a way you have a fascination for just general science because you've talked about physics and biology and the brain.

Torvalds: When I was younger I used to read a lot about physics and…

Booch: Who are your heroes from then?

Torvalds: And my heroes were apart from my grandfather probably, I mean, Newton, even though he was apparently not a very nice person, he was certainly one of those people who changed the world in many different ways. And, I mean, Einstein obviously, so that was physics. And I mean, I wasn't that aware of Darwin as a big mind because in Finland I was never-- I mean, that was never in doubt, no. This whole American school system is kind of broken.

Booch: You mean they don't teach intelligent design in Finland?

Torvalds: They don't, yeah, no. And—but I ended up being slightly-- I noticed when I actually did physics at a university that I wasn't as interested in it as I thought I had been. And it wasn't-- it just wasn't as interesting. It probably needed a bigger effort than I put into. So, besides being more about psychology and brain development and evolution and things like that, that I actually feel-- if I read some non-fiction books these days, it tends to be about human behavior or evolution.

Booch: Like what's the last non-fiction book that comes to mind that you've read or the last fiction book for that matter that you've read?

Torvalds: Well, I guess the last non-fiction was actually Michael Pollan's *In Defense of Food*, actually. So, that didn't actually fit that particular pattern at all. But it might have been *Oxygen* by-- I forget who the author. But it's about, well, at least some of the theories about how the presence of oxygen actually impacted evolution on earth.

Booch: So, next to the last question is out of total curiosity. Not counting spam, how much email do you process on a daily basis?

Torvalds: It's not that bad actually. So, A, I don't get any spam because my spam filters are extremely aggressive so. I mean, I get a few pieces of spam, but it's really not even noticeable. And I'm on a few lists, but they all go to a separate mailbox. So, my mailing lists end up being things that I look up if I need to, but I don't normally read them at all. So, the only email I get are literally emails that are sent to me personally, and I get too much of that. But I found that not answering email that is unnecessary is a great way to kind of push back the flow of it. Because it turns out that the people who really need to contact me either can just repeat the email or their question in the email was interesting and relevant enough that I actually answered it in the first place. It does mean that I actually, like, read a lot more email than I answer because my answering email is kind of-- it's a failure when I answer emails. Well, not always, I mean. Quite often it's because I'm the person who needs to answer that email, but it's something I've actually tried actively not to do.

Booch: Are you an iPhone carrier?

Torvalds: No, I hate cell phones.

Booch: You hate cell phones.

Torvalds: My wife has an iPhone, but I hate phones in general. I think they're a disturbing-- or like, email is so much better. And if you have to have a phone, you should just send SMS messages. But that doesn't work so well in the U.S. because it's too expensive and not everybody can take them anyway, right, there either. So, I think phones are a horrible distraction. And part of it is the synchronous nature of it, so you had to kind of-- I'm in the middle of something and the phone rings, and I hate it. The whole switching between two tasks. I'm good at concentrating, but the downside of that is when something disturbs my concentration, I really hate it and phones do that. Cell phones are just phones that are designed to be even more annoying. That's probably one of the reasons I had to move away from Finland, because if I hadn't moved away they'd probably deport me because Nokia is-- the whole Finnish economy revolves around cell phones almost.

Booch: And now the Lima [ph?] Foundation put Linux on the cell phones.

Torvalds: Yeah, well, yeah. I feel a bit bad about that.

Booch: Why do you feel bad about that?

Torvalds: No, I'm kidding. I wouldn't use…

Booch: The elements of your own destruction.

Torvalds: I can avoid the cell phones. If other people want to use cell phones, say, it's all good as long as I don't have to.

Booch: So, the last question is are there other questions I should have asked you?

Torvalds: No. No, I don't know.

Booch: Well, thank you very much for your time. You are a fascinating individual. And most importantly you're having a hell of a lot of fun and I envy that.

Torvalds: Yeah, it's a good life. <laughs> I am not complaining. Yeah. It's a…

Booch: Well, thank you very much.

Torvalds: Thanks.

END OF INTERVIEW