An Interview with

GLENN HENRY

OH 355

Conducted by Philip L. Frana

on

7 August 2001

Austin, Texas

Charles Babbage Institute
Center for the History of Information Processing
University of Minnesota, Minneapolis

Glenn Henry Interview

7 August 2001

Abstract

Glenn Henry begins by briefly outlining his education and early work career prior to joining IBM. The majority of the interview focuses on Henry's work as a software systems engineer and manager at IBM in the 1970s. Henry was centrally involved in the software development and hardware definitions for IBM's early Midrange Series computers. He had managerial roles at IBM San Jose and IBM Boca Raton, before leading a large team at the home of IBM Midrange Series hardware and software development, IBM Rochester. He details the programming effort for the operating system on System 3, the incremental advance with System 32, and the quantum leap IBM took with the development of System 38, a system Henry proposed – and one where the software was largely driving the definition of the hardware. The interview is particularly rich in detailing the relationship between software and hardware development as well as the technical and managerial successes and challenges with System 38.

This is an oral history with Glenn Henry in Austin, Texas, on August 7, 2001, conducted under the auspices of the National Science Foundation.

Frana: Thanks for meeting with me Glenn. You said on the phone that your work at IBM was an important story that needed to be told. Where exactly does that story begin? What education and experiences guided you when you were young?

Henry: I'll tell you very quickly. I started going to college when I was very young, at age fifteen. It was part of the Sputnik Acceleration Program and I went to the University of California, which is right next door to where I grew up. I remember seeing an IBM 650 computer there. This was in 1958, this system fascinated me. But anyway, a long story, after three years I dropped out of the university and flew helicopters which was my love at that time. I worked as a helicopter pilot at nineteen. For a variety of reasons, I quit that and went to work at Shell Development as a lab technician in 1973. The week I arrived, they got a new computer, an IBM 7094. And no one there knew how to do anything with it. So I was appointed as the junior flunkie to go figure it out. So in six months I was sort of running the place. No one could get any work done except when they'd come to me. This was in the days of high priests of computers.

Frana: You were programming?

Henry: I was the only person who could program it, who knew how to make it work; it was all mysterious stuff to the physicists and chemists. But that was my real love, of laying hands-on. So I love computers, number one. But number two, I was actually interested in – it was fun to be a high priest – but I actually got interested at that time in making computers available to the masses. This was in 1963, and at that time it was conceived that sure computers are great for biophysicists or chemists, and they are great for national defense initiatives or whatever they were at that time. But no one, at least in my circle, conceived of them as just a tool for ordinary people. They were very hard to program. You programmed in Assembly language or FORTRAN. There was no input/output on the computer itself, et cetera. So, anyway, I went back to school at California State College, now named California State University, at Hayward. It is just one university in the California State University system. Right there in the Bay Area, and I got a masters degree in math, which really meant computer stuff. In those days there was no computer science. It was an unheard of art. Anyway, I got a masters degree in a year. During that year I also worked for a paper company, and I worked on the problem of how they could control a paper mill – which is a mile long machine – using computers. So I got very interested in real-time control systems, and IBM at San Jose was building a real-time control system and so I got a job there. At that time IBM had a bad image, you know, for new people coming out of college, at least in my circle.

Frana: As a monopolist or…?

Henry: No. No. As a place where all my friends said I'd have to wear suits and I'd have to wear a gray hat, you know.

Frana: The 'Big Blue' image? Yes, right.

Henry: But I was interested in small computers and the only place IBM worked on the development of small computers at that time was in San Jose. It was called Small Scientific and Real-Time Control Systems or something. So I got a job there and I was very lucky to come in at the right time and at the right department. Within six months I was the lead programmer on our new small computer that they were developing. So my interest was on small things that real people could use. By real people, not just two hundred – you know the number of people who service big mainframes has probably only doubled since 1963, and it's a very small number compared to the potential users. That program was transferred to Boca Raton, Florida. My team and I were a part of that transfer and I was promoted to manager of the department. This is one of the classic stories. I was the lead programmer in the department, so they made me the manager. So I was a manager two years after joining IBM with no interest, no background, no training in management. I was just the sharpest programmer. But anyway, I was the manager and we were building a small, human-oriented computer. And that was transferred to Boca Raton and we were one of the first groups to go to Boca Raton. The facility hadn't even been built. This was a new IBM facility, which later became quite important in personal computing. So we did that computer which I'll mention in passing, because it was the first, I claim – with bias of course – personal computer. It ended up being a thing called the IBM System 3 Model Six BASIC. The hardware was a commercial system designed in Rochester. [It was] a small commercial system called the IBM System 3 Model Six. There was a whole family of these things. It was designed for billing and accounts receivable. It was a punch card oriented thing. A ninety-six-column punch card, which was developed in Rochester. As an afterthought they put a disk on it. We took that thing and we put an interactive BASIC program on it, we put a keyboard and a CRT monitor on it. It was as big as this desk. It had a removable disk drive, it had a keyboard, it had a CRT and it had a printer and it ran BASIC – Interactive Basic. And it was designed for one user to sit there. So I claim – this was 1971 – that it was the first personal computer. It was a little big [laughter], but that was what it was really aimed at. In graduate school I'd worked on a time-sharing system using both FORTRAN and BASIC, and I said, 'Well this is great, this is easy' – back to my theme of getting ordinary people to use it. The problem, of course, was the hardware cost was ridiculously high.

Frana: What did it cost a customer?

Henry: Probably twenty thousand dollars.

Frana: Okay, way more than a clerks' salary.

Henry: Yes. But there was a technical problem that interested me. And this actually has a theme that will lead into the main projects I did, which was, the hardware was designed by hardware people with no thought of the software at all. And then later, some software guys like us show up. In fact, we were trying to use another computer – a follow on to the IBM 1130. The IBM 1130 was a small scientific computer designed around SHARE and that was the original project we had in San Jose because that's where the 1130 came from. IBM in its mysterious ways said, 'Well look, we don't need two small hardware development projects. Rochester's building the hardware, why don't you guys use this?' So I remember going to Rochester, and I met Don Castella for the first time, who later worked for me on the System/38, and discovered that this hardware that IBM had enlisted us to work on had no multiply, you know, it was a commercial thing. It was string-oriented in memory. It had no registers. This was the System 3 architecture – it was just a disaster for what we were trying to do. We were trying to build a BASIC scientific computer. So one of the observations that struck my mind is that the IBM way of doing things is a bunch of engineers who came from, in those days, literally, the punch card era, would get together and build a computer. And various software groups would be applied to that computer. And there was no concept of a system. You know, hardware built for software. This was 1971 in Boca Raton when we finished the software packet. We were just doing the software; it was really a misfit. We had 8K of memory.

Frana: Now this was after unbundling, so it was sold separately then?

Henry: Yes. Well, unbundling happened in 1971. I was involved in some task force and it happened at roughly the same time.

Frana: The same time?

Henry: Plus or minus six months.

Frana: But did that influence the decision to work on these separately?

Henry: No.

Frana: No.

Henry: That was just the way IBM was doing it. So it was a crazy thing. It was a really nice software package it ran an 8K of memory, it had a software virtual memory disk, but the hardware was just atrocious. We just – I mean it was designed to run – the original hardware; the System 3 was designed specifically to be a 407 emulator. 407 is a punch card printer. It reads cards and has a punch panel and then it prints.

Frana: Could you run multiple batch streams even on the System 3 originally?

Henry: Not the original software. Later you could. There were several lines that were System 3; the System 3 family went on and on but it was hardware limited. And the original software was all punch card-oriented and then they said, 'Well, instead of punch cards we have a disk.' But the disk looked like a giant pile or tray of cards. But later, yes, you could run multiple things on it. It just sort of evolved. So when we finished the System 3 BASIC, I was sort of a star. You know I was getting a promotion every six months, and management liked me. I had a great deal of visibility, so they said, 'What do you want to do?' And I proposed that we design a new piece of hardware and a new operating system together. That we build a System, which seems trivial to think of now, but that wasn’t the way it was done there. Software as a discipline was very new to IBM. The original software was done by engineers. They just wrote simple little programs.

Frana: When did you first hear people refer to it even as software?

Henry: Well, I don’t know. But I think it was in the 60’s. I got out of school with my masters in ’67. I think it was just about at that time that people thought of themselves as professional programmers.

Frana: Yes, that sounds about right.

Henry: And inside of IBM the power was held by people that had been with IBM for twenty or thirty years that were familiar with the punch cards. Yes, they knew there was this software thing but they rarely thought of it. I really met people that thought programming meant moving the cords on this patch panel. There really was a physical patch panel you connected, this wire to that wire.

Frana: Programs were people.

Henry: About this time, because I was a rising star – I was in Boca, this is the ’70, ’71 time frame – I got assigned to a task force. IBM does this a lot, forms a task force to go review something. It was an audit of another project in Germany. So this was an eye-opening thing for me to see more of the real world there. Also, it made me lots of contacts, furthered my reputation. Anyway, so I started a project. I was the lead manager on it. I did the thing, came out and I did the proposals. The software people worked for me. The hardware people worked for someone else, but they were taking direction from me. And it was small, there were maybe ten or fifteen people, and we were going to build a new computer. It was designed to be a very small System 360. Now I have to back up. You know what a System 360 is?

Frana: Yes, sure.

Henry: So in those days IBM's mainstream was a System 360, and we were off on the side. And Rochester and Boca were off on the side building specialty, small computers. So I said, 'Why don't we build a small 360? And that would be an interesting hardware thing.' But the software challenge was the great one. The software for the 360, now and certainly then, is just terrific. It's a batch-oriented operating system right, with JCL control language. There was no concept of keyboard CRT's or time-sharing systems in those days. So we said, 'Let's build a new piece of software that's oriented toward a single user.' All right, so we started that project. And it was hard and we had fifteen or twenty people and we had charts, and I built the spec and I sent it around. Again, it's a typical IBM story. They called me and said, 'In our wisdom, we are going to send you and your project and the people who choose to go, to Rochester, Minnesota.' And I said, 'Well I'm not going to be doing that.' And they said, 'Well, Rochester is building this new piece of hardware and they figured your software and their hardware could go together.' So the new piece of hardware they were building was a small, sort of single-user, but it was a single operator – it was a commercial system called the System 32. So off we go to Rochester. And I became the manager of the operating system for this hunk of hardware that had again been predesigned by some guys in the corner called the System 32. It was designed to be a small version of the System 3. System 3 – System 32.

Frana: Was there reluctance to leave Boca Raton, on the part of your team?

Henry: Yes, in fact my key engineer who was in true life a very good friend of mine, refused to go. We had come from California, and we had gone to school in the 60's, all of us in California. And [we had] come from California to Boca Raton. And Rochester looked like the end of the intellectual world, as well as the weather – sorry to tell you that as you are from Minnesota – but it didn't look attractive to us. There was no university in Rochester, the weather was cold. It was obviously a different sort of intellectual culture than in the Bay area that we had all grown up in. Our team had all started back in San Jose. We went to school there and [had] wandered this way. So I lost about half of the people. There is another interesting story which has nothing to do with it, but I'll tell it anyway. I met my wife, who is a programmer at IBM, in Boca Raton. So when they called me in to say we are going to move you and your project, I didn't hesitate because I wanted to do this project. All right, I was young whatever, right. And I said, 'But I am engaged to be married to my wife over there.' And they said, 'Oh, no problem. If she's an IBMer we'll find her a good job in Rochester.'

Frana: And she was from Florida?

Henry: Yes. And then they said, ‘By the way, we figure you ought to be able to pull this project off in two years, so this is, if you want, a two-year assignment. Then if you want, you can come back.’ Those were two of the great lies, along with the check is in the mail. We’ll find a good job for your wife and it’s a two-year assignment. [laughter] So off we go and we get married on the way. We have a honeymoon. We show up in Rochester. And I show up and they’ve got a team ready and people are eagerly waiting for me. And, of course, my wife comes in the same day and they say, ‘Who are you?’ What? You know, no job, right until she gets assigned it. It was a terrible job and that later became a great strain that got us out of Rochester. But anyway, that’s off the point. So, I was managing the System 32 operating system, which included the languages – the operating system and the compilers that go on top of it. And it quickly turned into just a commercial system. It was a good little system but it was designed for commercial applications. So I basically said, ‘Look at it, if I’m in Rome, I’ll speak Italian.’ And I said, ‘I am going to give up on the small scientific stuff and I’m going to focus on small usable systems that do commercial applications, because commercial applications are the bread and butter business applications.’ And if we can do those easier, et cetera. Again though, I was struck immediately by the problem that the hardware and the software were gross misfits. The hardware was designed just in the back room and it was designed solely from a cost viewpoint. They said, ‘Hey, here’s the System 3, it costs so much, how do we reduce the cost?’ Well, the things you do to reduce cost often are the wrong things to do from a software viewpoint.

Frana: Yes.

Henry: I mean, maybe there are other things you can do to reduce costs. Obviously you need to reduce costs, but it was a very poor fit. So immediately…

Frana: So this was despite the fact that, Herb Pelnar, I remember said that the System 32 was one of the few times where the hardware and software sides really were going on at the same time, rather than it being sort of layered.

Henry: Well, what happened is we tried to change it. In fact, we did – here's another first at IBM and maybe our industry. One of the ways they decided to reduce hardware is they would not hardwire the system's instruction set. They would microprogram it. Do you understand the concept there?

Frana: Sure.

Henry: Microcode?

Frana: Yes.

Henry: So I got there and I said, 'Aha, the computer really has the real hardware instructional set isn't a System 3 at all,' and magically the real hardware instruction set had registers and things that were interesting to software people. So I proposed, and I believe it was the first time, we move portions of the operating system in the microcode. And I wrote those portions myself. I remember sitting at home writing code, you know. I took over a room, stayed home, you know classic thing – worked night and day on code. And we moved the, basically the disk subsystem, which was very critical in the microcode. So I was out on the hardware myself. I was, at this time, a second manager because I was hands-on, okay, I still am. So the core things about the hardware were frozen. But we were able to influence it because it wasn't finished and we were able to dive into microcode. Fortunately, the engineering manager at that time, a person by the name of Keith Slack had come with us from Boca Raton and he was on the team. He was one of the guys working on my project in Boca Raton. He later became an important person in IBM. In fact, he came from San Jose. So he was an engineer on the small scientific computer, he had gone to Boca and he was sort of the lead engineer on the small 360 I was trying to do and then he was sent to Rochester and ended up only in the CPU portion. It was the CPU, the processor engineer. And so he was amenable, he knew me and was amenable to my concepts that we should marry hardware and software together. So they were designed differently. We were given the System 32 as a concept. But it wasn't finished in one part, it was microcoded. So software burrowed into microcode. And this is the first time I know of that we did that type of thing. We actually moved what anyone would consider real operating system function, you know Beta management functions into microcode interwoven with the System 3 emulator.

Frana: It really is an impressive achievement.

Henry: That was 1973 that we did that. I believe that this is ’72 and ’73. I went to Boca Raton in ’71, so during the period ’71 to ’73 I was managing the software for the System 32. But immediately I started proposing a brand new system. This was to become the System 38. This was my dream. I said, ‘Let’s start with a clean sheet of paper, all right, and build hardware and software together, and build a system that is a quantum leap beyond the entire System 3 thing.’ This was a very controversial idea partly because I had to point out the deficiencies, in my mind, that existed in System 3, which is what Rochester has built. So I am the new guy from Boca Raton saying, ‘Look at this piece, why don’t we build a system that does it right.’ I didn’t put it that way, obviously, but that was part of the politics. The other part of the politics was this was software-driving hardware. That’s what the System 38 became. My team and I drove the hardware definition in every aspect of it. And this is again, I think, one of the first times, one of the fewest times at IBM that it was done that way. And that was also controversial because the management was all hardware people. The lab director at that time was the manager of the group that built the 96-column card machine. It was just a different era, a different discipline.

Frana: So you actually commandeered engineers to become programmers then?

Henry: No. No. We used programmers for programming but we did architecture. Which again this seems like a silly word, now everyone does computer architecture. But computer architecture wasn’t a word then. What we did on the System 38 was very interesting. But let me back up. We had a unique opportunity. During the period of ’71-

’73, my official job was managing the System 32 software. But I was proposing this and management, either because they believed or to get me off of their back, said, ‘Okay, take nine or ten people and you work on it on the side.’ So I had two jobs. But because we couldn’t really do anything yet, I didn’t start it full time and start building a big team until ’73, we were able to do it right. And right in my opinion, we started with first principles. What did we want the function to be? What did we want the user interface to be? What did we want this thing to achieve? What were our goals? Things that sound very basic, top down things. And then we derived from that to that to that. So for example, one of our great inventions was a single-level store. And I will talk more about that. But we didn’t start off saying we want a 48 bit address. But hardware ended up having 48 bit of addressing. Real hardware addresses, 48 bits, which was just unheard of in those days. We started off saying, ‘Here are the kinds of applications we want to run. We want to make those easy we want to integrate tightly a database – a relational database.’ That was another invention.

Frana: Now does that come from San Jose?

Henry: Yes, the relational database. But we were the first to build it into the operating system. It was previously an add on, tacked on.

Frana: Entirely new?

Henry: Yes. The concept was developed in San Jose research. At that time, the operating systems for the 360's, 370's I guess had been introduced by that time, they had a key orientated kind of database system. Quite different, in fact, the physical disk was not set for sectors like we think of today, instead physical keys on the disk. And so they were building a relational database on top of this in San Jose. We took the concept and just literally built it into the core of the operating system. We extended the RPG language to RPG III, that was done by us. We said, 'What you need is a relational database built into the language. You need to run that efficiently so it looks to us like a single level storage as opposed to a disk and virtual memory would be appropriate. And what do you need to support that? Well, we need 48 bits.'

Frana: And where had you heard of these ideas for virtual memory and demand paging?

Henry: Well, one of the task forces I worked on back in Boca Raton in 1970, I was assigned again as a junior hotshot to a task force that was studying applying virtual memory to small computers. IBM had developed virtual memory or had first implemented it on a 360 that was scientific oriented in about 1967.

Frana: That sounds about right.

Henry: Again, I was on two or three audits in Germany. At that time Germany was developing the low end of the System 360 hardware line. And again, we had this schism. This is still true, at least it was when I left, hardware was developed here and then the common operating systems were developed by different groups. So you had MBS and a low-end system, which was called DOS. Not the same DOS we know in pc's. But anyway, so Germany was developing the smallest, at that time, System 360 hardware and we were sent on a task force to study what the software should be and one of the key questions was – virtual memory was considered an option, it wasn't until probably '73 or so that IBM mainframes actually supported it. But, it turned out that I had become somewhat of an expert, in those days, on virtual memory, because remember the System 3 Model Six BASIC, the first personal computer that we finished in 1970 in Boca Raton. That was the thing I worked on. I told you it was a System 3 with BASIC. The thing only had 8K of memory so we actually implemented in virtual memory. But the managing **[?]** the demand page and everything was done by software, since BASIC is an interpretive language. Our interpreter simulated a much bigger memory and did paging and everything totally in the interpreter. So in the process, I had written a paper that was published in '69, I guess, in the IBM journal. This is going way back, I forgot this. Before I got to Boca back in San Jose, I had got an engineer to build or an engineer had built, I really don't remember, a kludge on the IBM 1130 that would do simple address translation. I wrote page and routines. And another person and I wrote a paper which was published in the IBM, some IBM software symposium journals in '67-'68-'69. I don't remember that time frame. And it was the classic kind of paper you still see which are curls of page **[?]** sizes, right versus various replacement algorithms. And we had done this on an 1130 kludge, with just a hardware kludge and we had written all the software to turn it into a paging machine just as an ad-hoc experiment. So I had established my name as a guy who believed in virtual memory and had done research. That was the first thing. Then we had implemented virtual memory on our little System 3 BASIC. And so I knew something about it. Now, virtual memory is actually pretty simple, so you don't have to know too much to be an expert, but in 1971 if you knew anything you were an expert. In 1971 as I remember, when the System 370 came out, that was the second generation System 360's, the high-end systems had virtual memory on them for the first time. So anyway, I was a "virtual memory expert" and I developed it more by studying the low-end 360's on these audits. So virtual memory was just *prima fascia* assumed by me. I was a big strong advocate of virtual memory. I actually gave talks on it. But we ended up going way beyond it. The point I was making in my original rambling was that we at Boca sat, back into the '71-'72 time frame, and thought top down, starting with what is the function we want to provide to the user? What are the languages and the operating system interface? What is the operating system API, though the term API didn't exist in those days – how do we want to structure the software? And again, it evolved from me spending all my time and energy in System 32 software so that by 1973 I was spending about twenty percent of my time on the System 32 even though I was a manager of about fifty-sixty people. I was just personally working on this other thing.

Frana: Did you envision who your customers would be? Was 3M involved in this at that point?

Henry: No. Our customers were not – we didn't envision specific people – our customers were small businesses that wanted to do billing, inventory control, account and receivable sales analysis. They wanted to have interactive versions of those, that was very key. They wanted to do things like query databases. We saw that coming. But we saw this as a replacement to the small business line at that time. So by this time I had completely given up my interest in small scientific systems and said, ‘Okay we are going to build the best small business oriented system in the world.’ So among other things, we studied the System 3 operating system at the time and the kind of applications that came in there. And one of the important things we did, which I believe again is somewhat controversial, but I believed in it, we developed what I call the – and there were six or eight of them – the ‘critical precepts,’ the things that we would never trade off. And because one of my beliefs is you have got to have precepts in computing and you don’t trade those off. And that later, of course, became a problem because System 38 was much harder, it took much longer than we thought.

Frana: Yes.

Henry: But I wouldn’t give up on my core beliefs even if people tried to change them I said no take the time until we do it right. And basically later I forced people to say either you replace me – at that time I was a big manager of six hundred people – or you do it my way. And there was a serious debate over that. In fact, I was involved in it. But we had developed certain core beliefs that we said if you are going to do a total from scratch system these are things we want to do. There was no hardware at all, so we envisioned completely new hardware. We were defining what the hardware should look like, not the design of hardware but the function of the hardware, the instruction set, things like that during this time. So in 1973, they said, ‘Okay, we’ll start this project full time. You are the architecture and software manager Glenn.' This other old grizzled veteran Ray Kloss was the hardware manager but he had no concept of software architecture but he didn't need any. He was just an old-timer who managed engineers. And we needed a systems manager, who's really an IBM systems in terms of business, manager to run this. So a guy I knew, Brian Utley, was brought in from Boca Raton. He had gone to Boca Raton from San Jose. This was this whole crowd coming back together. The architecture design, the hardware instruction set, was defined by my group, but the hardware guys actually implemented it. So we defined that instruction set and what would now be called the computer architecture. And then we implemented the operating system and the languages, because for the kind of system we were building the languages are tightly related.

Frana: In what languages – I mean RPG was a nonprocedural …?

Henry: RPG, COBOL…

Frana: And then COBOL. PL/I?

Henry: No.

Frana: FORTRAN?

Henry: We did a FORTRAN, yes. But RPG and COBOL were the two big ones – those were the big commercial languages. RPG, which was extended to be RPG III right…

Frana: That's Tom Hoag?

Henry: The manager of the group was a person by the name of Dick Sulak.

Frana: Okay.

Henry: Tom Hoag is a name I remember, but there were lots of people involved in that. Because Rochester developed RPG basically as a language. But we extended it substantially by adding database and interactive features directly into the language. So reporting to me, I sort of have four – I can't even remember the organization – but we had a language and utilities group, and Dick Sulak was the manager of that, you know the operating systems group. I had a VRM group that were going to come and talk about that, basically that was the core of the operating system. The operating system was divided into two pieces. I had a test group that was Dick Hedger. And after we got started Sulak joined the team and he was originally a submanager within the guts of the operating system and then became manager later. You know that group.

Frana: Yes. Don van Ryn, was he there?

Henry: He was. He didn't directly work for me; he was in there somewhere.

Frana: And Roger Taylor?

Henry: Roger Taylor worked directly for me as the manager – we need to explain the split of the high-level operating system – and later he was removed from his job at the end when we got in trouble. There was a great deal of controversy involving that.

Frana: I'm hoping that we can talk a little bit about that.

Henry: Okay. But anyway, so in 1973 the bell rang. I was full time on System 38, right, gathering up people. The engineering team was gathering up people, there were meetings, charts, and pictures and off we went. I should stop rambling for a second. Probe me with a question.

Frana: You are doing very well, your memory is much better than …

Henry: Oh, let's talk about, at this time IBM …

Frana: System 34 – are you working on that simultaneously?

Henry: No.

Frana: That's just an evolutionary step?

Henry: That’s an evolution of the System 3.

Frana: Okay.

Henry: And there was a great deal of strain. IBM intentionally creates competition within itself, there’s sort of a Darwinian thing. The System 34 was a hack of the System 3. It was an evolutionary improvement of hardware and software. And that was being developed during the ’74 and ’75 timeframe. And that was way ahead of us because it was a much simpler project.

Frana: Sure.

Henry: So we immediately saw ourselves as a replacement to System 34. But there was definitely competition between the two groups. We perceived it as this old tired clunker that is missing two or three pistons, right, the wheels are flat. They perceived it as this other thing that is actually working and we are this pie in the sky dreamers in the ivory towers. Because our stuff looked fairly dreamlike to most people at the time. The other thing that happened in the ’74 to ’75 timeframe is the main IBM, at that time, had decided they were going to replace 360’s and 370’s with the new family of hardware and software called FS – that’s for Future Systems. That was a huge deal. I mean, in typical IBM fashion within a year they had thousands of people working on it. Well someone noticed that the top-level bullets on the features of FS were, for the same reasons, the same as ours.

Frana: These were part of your six core precepts?

Henry: Yes. The top-level pitches looked alike. We, of course, had developed ours totally independently, we didn't even know about FS for a long time. So IBM created a task force to see if we really should be killed and if the low-end of FS should replace us. So I went to look at what FS was, I got their documents. I said oh my god, I mean FS was a joke, it could never be built. This was like Star Wars in 1970. It wasn't going to work and any fool could see it except the thousands of people and IBM management couldn't see it. So one of the little known and little understood things, one of the best things I ever did was protect the System 38 from being killed in '73 and '74 by FS. By just stalling, waiting for the inevitable to occur, which is waiting until someone noticed FS had no clothes. And they killed the thing. It was a great disaster because they hadn't been developing a backup system right, heads rolling. Now I don't know if this story has ever been told. This was a big deal. They were going to replace all their systems. And they killed it and when they killed it there we sat right untouched out in Rochester just grinding along. So we were doing two things. I was putting up a fight, a front, smoke, whatever it took to hold off the forces of IBM while we just kept building underneath. So when FS died it didn't take us with it. We were perceived to be small enough and simple enough. So I was just always billing us as a bunch of hicks out in the cornfields of Rochester building this toy, don't worry about us. We tried not to get associated with the thing. Anyway, that was just another passing thing. I’m remembering now all these things at the time seemed like life or death things. I was traveling to Poughkeepsie all the time and to Harrison, New York, you know all the standard IBM places to argue why our thing was different and not only different, but better. We should be untouched by this mess.

Frana: Whom were you reporting to at that time? Was Martin Belsky a player at that point?

Henry: No.

Frana: Was Watts Humphrey?

Henry: Those were all ‘the Enemy’ at that time. They were all on the big systems. Those were big systems guys. I reported to Brian Utley, System Manager 38 who reported to Harry Tasjian, the lab director of Rochester. Harry Tasjian was the old card punch manager.

Frana: And he’s no longer with us? He’s no longer alive?

Henry: Is that right? I don’t know. He was a very interesting guy. He would beat me up every day walking around saying, ‘Glenn, I think I may replace you today and I don’t think this project is going to fly,’ but by the end of the day he’d be supporting the project.

Frana: He wanted you to convince him? On a daily basis?

Henry: Yes, I hate to say ‘the Enemy’ but inside of IBM, I mean it becomes that way. The forces of mainframes to us were the enemy because they were constantly talking about how they were going to build little baby things and take their line down. And we thought that was foolish from protectional reasons, but it was also a problem.

[tape 1 side 2]

We looked down on the way they did it, which was totally separate. They had huge programming labs, thousands of people. Completely in it from the hardware development labs, right, because they had this concept, they had this well-defined hardware interface, which is instruction set architecture from System 370. So the hardware could proceed and the software could proceed in these. We were of the System that you need to do these things together right and you don’t need thousands of people.

Frana: So they really weren’t learning from Frederick Brooks?

Henry: In fact, the *Mythical Man-Month* was one of my favorite books. I’d pass it out to managers and things.

Frana: It’s been reprinted many of times.

Henry: Yes. No, IBM didn’t learn. In fact, later I finally, as an IBM Fellow, which is like a job you never leave right, I left IBM in disgust because they were just screwing up so bad. But anyway, that’s another long story. *The Mythical Man-Month* was very appropriate because we fell prey to one of his most basic things. One of the things he said is the most dangerous system is a second system. Because you try to fix all the things. You now think you are smart enough you can fix all the things you didn’t do right in the first system. But you are not mature enough to know that you can’t do all those things. So there’s sort of good news and bad news. The System 38 was underestimated in terms of its complexity significantly. I think it’s one of the great – I’m biased of course – but I think it’s one of the great technical advances in operating systems, certainly within IBM. On the other hand, if we hadn’t been young and idealistic, it would have never got done. We were beaten upon from every side. I was technically oriented. Originally I had developed all the charts, developed the key concepts. But after a while my key contribution was being a good salesman and a good person working with upper management to keep us alive, to keep them off our back. This was not a lock, it was a project, and it was trying to be killed every month. Never for the right reasons. If anyone had ever shown up and really explained to us how we had underestimated the schedule, but that wasn’t why they tried to kill us. It was for political reasons, sales reasons, it was conflicting with the 370, it was conflicting with FS, and it was conflicting with the System 34, those types of things. The few technical audits we had with people collected from research and the other programming labs. And in those audits, as I look back on it, there were snippets of wisdom, where they said this is really going to be hard to do. The problem is that that wisdom is delivered in a political context, you know. These are guys coming from the mainframe systems trying to kill us. And so naturally it's ignored because it is part of a political message, not a technical message.

Frana: It is not as if 360 and 370 didn't miss all sorts of deadlines.

Henry: Yes. Yes. Lots of them. But we did fall prey to the second system syndrome and underestimate the complexity or over specified, but I don't think we over specified. I believe the key concepts became the core of the AS400, which is a successful system. What we did, basically, is we outstripped the hardware of our time. The software capabilities required too much hardware to be cost effective in 1978-79.

Frana: So keeping the price down was very difficult?

Henry: Yes.

Frana: It could have been done?

Henry: Yes. I remember going through, 'Okay, it has got to work in 64K,' and then I said, 'Well, it won't work in 64K so 120K, well maybe we need 192 for performance.' Those were big deals in those days. So later when hardware became cheaper, these principals, now I believe are being seen as very good principals, but there was a strain at the time because we were driving the hardware cost up and we wouldn't give up. The biggest single technical thing that caused the hardware cost to go up was probably our single-level store and the amount of code we were doing. The biggest controversy, of course, was the split of the operating system into two pieces. One was called the VRM microcode. Are you familiar with that?

Frana: Yes.

Henry: That was a fistfight controversy.

Frana: How was that decision made?

Henry: I made that.

Frana: You made that decision.

Henry: That was my pet project. I just rammed it through. And that's the one thing, I don't know is right or not; it's one of those things you can never know. The user-oriented principals, in a way relational database, high security – that was another thing, no operating system had any security in those days. We were the first operating system to have security as one of our five basic precepts. And we tried to put security all over this thing. Now using addresses, tag pointers, these were really computer science concepts. But that was our goal. The technical things like that, I believe, are fundamentally good and we were just ahead of our time. And luckily, we were able to keep the program alive until the time came. The decision on the VRM was not made for technical reasons. It was a pragmatic decision that I believed in and you can’t have it both ways. I decided that was what we were going to do and once we got started on it I didn’t feel we could turn back. And so I ran over, unfortunately, everyone who got in the way of it. But we either had to do it or not do it once we had done it for three years, right?

Frana: Yes.

Henry: To turn back would have cost us very high. The basic idea was – I guess it was made for technical reasons also. This goes back to my System 32 days, where we moved hunks of the operating system into microcode. There were two basic premises why we wanted to do it. The first reason was as time went on the hardware would get more powerful, right. And one of the things that was clear even then was that you can never change instruction sets for hundreds of thousands, or what became millions of lines of code operating system. You can’t today – to this day the 360’s are running millions of lines of code of a similar language. So I said, even at that time it was clear, so I said, ‘We must eliminate dependencies upon the initial hardware because the initial hardware, if we are successful, will change and it will change and it will change, right. See what was hanging over my mind, this is why it is a technical decision this piece of it, was the 360 environment. Even in that day, in the mid 70’s, the 360 architecture was obsolete, but it was clear they were stuck with it forever. Which they are. So I said, ‘We can’t let that happen to it. If we are not successful business wise it doesn’t matter what we do. If we are successful the hardware is going to undergo this 2X per year kind of evolution right?’ No. But anyway it is going to evolve in a lot of ways we can’t even imagine. And we can’t have lots of code and applications that, you know, know about instruction sets, that hardware. So I said, ‘What we must have is a high-level machine interface. We must say that the instruction set of the machine is here, even though today’s hardware is here and we will bridge that with microcode.’ And I said, ‘There’s no point in defining a high-level machine interface that looks just like another hardware instruction set.’ Those are the things that become obsolete. Because at the time our hardware couldn’t even do what a System 360 could do. So we could have put a 360 interface here. So I said, “Well, what we’ll make that interface like is a functional interface.’ Sort of like an operating system interface where you call functions, and key things like disk management, virtual memory, and tasking. That allows us total freedom in reoptimizing those in hardware. I kept going back to my previous project where I moved the disk management system, if you will, into microcode. So this gets back to our top down thing. We said, ‘Here are the things that are fundamental to an operating system: its managing physical memory devises, its managing IO, and its doing managing time – doing the time slicing whatever your task switching algorithm is – all those things will be done in “the hardware.” Now if our current hardware isn’t up to the task, fine, we’ll build code here. And that was called the VRM.

Frana: To bridge the gap. What was the debate then? I mean this sounds like a very good technical reason.

Henry: Okay there’s one other, let me give you the other reason too. The other reason was pragmatic. So that was the technical one that I believed in. The other reason was a pragmatic reason. I said, ‘We’ve got to develop all this software and all this hardware simultaneously and what we ought to do is to find an interface to allow people on either sides to work independently. So that if the hardware changes it effects just this much not the whole thing.' To me a very pragmatic way of starting from scratch developing things is you layer it with interfaces so people work against interfaces. And that interface isn't going to change. And that actually happened. The hardware, the actual instruction set of the process we use actually did change sometime along the way radically, but of course it didn't reflect up above the VRM. So I had a VRM group that knew about the hardware and then Roger Taylor's group was the, I forget what we call it – the operating system group – and they didn't know about the hardware at all. They just knew it as VRM. That was what we documented. That was the first thing we ever did. We had a giant speck of what that intermediate level was. I might have been confused about the name, I think it was the Virtual Machine Interface. So it was a pragmatic way of developing massive amounts of code and hardware in parallel without having a continuous ripple effect that stews the whole thing. Yes, it sounds like a good idea when I say it. I really believed it. It was controversial because it cost performance. There is a cost to performance going through implementing this interface versus just widening the operating system right down the hardware.

Frana: Right.

Henry: Cost to performance. I guess the other reason it was controversial was just individual people's philosophy and what have you.

Frana: Like a short-term view versus a long-term view of the system?

Henry: Yes. I don't know. I mean, it's just people's style of doing things right, you know? But the major argument was performance loss, which was valid. So I always, good or bad, the System 38 to me – I used to joke about it – in fact, was intended to be a ten to twenty year architecture which in those days is kind of like a long time. And that's what it evolved as different on the AS 400 but that's perfectly fine, right. I see the key elements in there and so the 38 worked and then the hardware changed and the languages change, evolve. But the key concepts are still there. So to me it was successful that way. I am not saying I am totally right, but at least what I was trying to think about, one of the architectural things you do to preserve it, because hanging over me was the disaster of the 360 where they had architected it. You see, the 360 architecture was brilliant in one sense, it was the first time in history someone had had an instruction set that was the same across different hardware. That was a major breakthrough.

Frana: Across the entire product line?

Henry: Right. But the problem was ten years later it was the wrong instructions.

[laughter]

Henry: I mean everyone knew it. That's why IBM itself was trying to get rid of it. They could never get rid of it. They could just keep enhancing it and acting it. A very similar thing happened in current business, we were building the Intel compatible microprocessor here and it’s the same strategy. It’s just that the architecture was a kludge that has got stuff in it that’s twenty years old. Because once you’ve put an instruction in there you can’t ever make it go away because something will stop working. We have instructions that were invented in the 8080, the 80’s are still living, or even before that. But anyway, that has always been one of my big themes, which is to try to make software less dependent. It’s a funny thing. I don’t know if it is funny, but I wanted to do two things. I wanted to design hardware and software together and then I wanted to make the software as independent of changes to the hardware with time as possible. And those two things said to me, this high-level machine interface, the VMI. And what was below that was technically microcode. It was written in a different language and done by different people working for me. And what was above it was an operating system.

Frana: And this was an objection the FS people had to this VRM?

Henry: No, I don’t think so. They had the same concept.

Frana: Oh, they did?

Henry: Yes, as I remember. Yes, they had a high-level machine interface. I believe any architect would come to a similar conclusion if you think about the big picture trying to build something that has a long life, right.

Frana: Especially with the 360.

Henry: It's what you are as a programmer down in the bowels of the operating system. You say, look if I've got to go through that tired guy's interface to get down there. It's a complaint today on any operating system, including Windows. People complain they've got to go through the Windows API. They can no longer directly get into protect mode and other things. By the way, do you want anything to drink? All I have here is Tab.

Frana: Sure that would be fine.

Henry: Do you know Tab?

Frana: I remember Tab. I'm not that young.

Henry: One of the last batches of Tab here.

Frana: I didn't realize it was no longer being manufactured.

Henry: Well, we've captured the market here. This is a joke. You talk to people in Rochester. I was also a Tab-aholic there. I was well known for it. Okay, probe me with something.

Frana: Now, was System 38 distributed in any of its capabilities?

Henry: No. That was a concept that showed up late in our design. We started learning about and reading about it, but in the original design that was not one of our concepts. Hardware was so expensive in those days. That dominated a lot of our thinking. There were no pc's. There were no pc's, there were no workstations. There were no things with keyboard CRT that had a brain in it. And networking as we know it today didn't exist. All the communication was either the star communication, the keyboard CRT's with the mainframe in the middle. You know TSS – time-sharing system. Point to point bi-sync. STLC, SNA, the classic mainframe.

Frana: So those were the only two things you had in mind. You weren't thinking distributive computing?

Henry: Yes, we did have terminals. They were the classic hardwired terminals like the 3270's. We used a different terminal but it was a keyboard CRT with the brains back in the controller. It was just the way all systems were. The hardware was very expensive and there were no microprocessors. You know, single chip microprocessors that could see, catch, and smell.

Frana: When was the original release of System 38 announced then?

Henry: Well that of course gets into another story. The manager of the project, Brian Utley is a very aggressive – he was a system manager. At that time he had a profit and loss responsibility for the project.

Frana: I'm sorry I didn't…

Henry: Meaning he had P&L responsibility for the project. He managed above me, but he also had marketing interface groups, product marketing finance groups. The system manager owned the system. And he'd be measured on the business success of the system. This is a very important job at IBM and IBM's development. He was an extremely aggressive person. And if you combine that with me also being aggressive and somewhat naïve about the complexity of the system. We decided to announce it very early. Today, at least in the business I am in, you don't announce until you are ready to ship or after you ship. Those days you announced ahead of time. You needed to do that because the sales reps, of course, the customers, everyone had to prepare, applications had to be written. So there was tremendous pressure to announce it.

Frana: Yes.

Henry: Okay, inside of IBM there was a check and balance group called System Assurance that has to approve announcements and shipments. A third party group – you know, IBM's watchdog.

Frana: To make sure that if they announce something they are able to deliver.

Henry: It was, or at least in those days, a very important part of the culture. System Assurance comes in, reviews. And there is a checklist of a million things you have to have before you announce. And basically again Brian told me to just run over them. My guys and I were quite good. We ran over them, bamboozled them with charts, pushed and I won't say tricked them into agreeing to announce, but it's hard to explain, at that time we thought announcing was an end unto itself. We sell it when we announce, we celebrate it, right. Yeah, we've announced, we dance. You know we thought System Assurance was the greater enemy; those guys were standing in the way of our getting to a tassel of great things. We'll, run them down right, overpower them intellectually.

Frana: And where are these guys from? Who are these people?

Henry: They were from the same laboratory. They are the second group that lives in the development laboratory. So, they are always looked down on by the development group. You know they are in the way. We're developers who want our product announcement, we want publicity. By this time we had been working on it forever. So it was announced in the fall of, I don't remember, August/September maybe of '78. Is that right?

Frana: That sounds about right.

Henry: Or was it ’79? I guess it was ’78. And we announced that it was going to ship basically a year later. We announced it in the fall of ’78 and we were going to ship it in the …

Frana: Fall of ’79?

Henry: Yes, the summer of ’79.

Frana: The summer of ’79.

Henry: Yes. See you announce it long before it is done. Well, not only *Mythical Man-Month*, but 10,000 other books and experiences of life says that the last ten percent is ninety percent of the work. That is definitely true in a big complex operating system because of all the interactions. This was a big thing. This thing had ballooned to 5 million lines of code by now, which in its day, our tools were grossly primitive versus the tools you have now for developing projects.

Frana: That’s huge! I mean the original OS/360 only had a million lines of code.

Henry: Right. Yes, I guess. So we had to go on into a classical big software project and the tools were primitive. We were probably one of the first development projects ever to use interactive. Before this project we’d always used punch cards, you know batches. Take your tray of cards and submit them. We had terminals, by DTSS, right, but the turnaround time was a day. You'd make a change, and if you'd make a syntax error or something in your program it was a day before you got feedback. You know, now it is seconds. But anyway, it's amazing that we actually got it done, when I think of the tools I have now, versus then.

Frana: But to a certain extent it was – well I guess it was pressure from Brian, but it was sort of a game. Well, it's going to be another year anyway, a year and a half.

Henry: Announcement was a goal right. And we are goal-oriented people, aggressive. And there was tremendous pressure from Brian. But I wanted to announce too. People did too. That was the goal, we've announced. Of course it isn't the goal, the real goal is to ship products and announcement actually gets in your way. You see, the way it is supposed to work, you are supposed to go through the review activities for announcement with System Assurance. You are supposed to shed light on the truth, right, whether or not you are ready. But since our goal was to announce, light wasn't shed on the truth. We convinced them we were ready to announce.

Frana: Yes.

Henry: Now, later they got in trouble for this. When we slipped, obviously, we had trouble, but they [the system assurance managers] got in worse trouble because they were the policemen who let the robbers run wild. You expect robbers to rob, but that's why we have policemen. So it became clear that we were going to miss the date and there was talk of a one-month slip, then it was a three-month slip. At this point, I and the senior manager of all Rochester, Hal Martin, applied another precept which is take no small slips. You don’t solve the problem a month at a time. You decide. You figure out what it is going to take to do it and you take that. Our system manager Brian did not like that concept, partly because at that time the pressure on him was severe. He was the senior manager and he had made promises, right. He was the guy accountable to the CMC, which was the Corporate Management Committee, headed by the CEO of IBM. It was that level decision. I’ll come to that in a second. But basically, Hal Martin, the lab director, who again unfortunately died very prematurely, was one of the saviors of the project. He personally took over decision-making and he came and saw me personally. He looked me in the eye and said, ‘Glenn you’ve got to tell me the truth. How long do you think it is going to take to do this?’ And I said, ‘I think we are going to slip a year, maybe nine months.’ And this was the time when people were saying a month to three months right. And Hal said, ‘All right, if that’s the way it is going to be, that’s the way it is going to be.’ And he overrode Brian and said this is what we are going to do. Now, at this point, it is in the summer of ’79 when we are supposed to be shipping, instead we are slipping a year. This is such a big deal, Brian has to go see the CMC you know. And I go along with him. I didn’t get to go in to see the CMC. But in the pre-meetings, it was very interesting. They had a number of executives in there, Brian, the lab director, Harry Tasjian, and Hal Martin who was head of all of Rochester, his boss, Dave Slattery, the vice-president of the headquarters of our group, and me and Brian in a room. This is getting ready to go into CMC. This is the final, ‘What do the charts say? Is it right?’ And Hal starts asking questions about, ‘How long is it going to take? What about this? Are you sure about this?' Brian starts answering and Hal says, 'Brian, quiet, I am asking Glenn.' So basically, I was put on the spot, I mean that was really the way it was. It was, 'Glenn, tell the truth now.' It wasn't just about schedule, it was about performance, it was about certain features which were to be made right.

Frana: They were starting to ask questions about other things?

Henry: What?

Frana: Because System Assurance had been snowed?

Henry: Yes. So this was, I felt at the end they were asking me as the manager and the inventor, you know the hot-shot technical guy who was getting gray very quickly here. But I knew the product. I was the guy driving it. What I felt, what I felt I could commit to, right, and they were going around me and I gave them the answers and they said, 'Okay'. And they all traipsed off to that level of people leaving me there, to see the CMC, which is the Corporate Management Committee, which is the top three or four managers of IBM – the group that has always had the head of IBM in it. And Brian is a classic example, would never talk, pass down problems he had right. But I heard from other people that they just really read him the riot act. Beat him up miserably and basically after we shipped, sent him off to the penalty box. He later came back, but he was sent off to be a marketing manager in England or something. That was the IBM style in those days. But apparently, I think they beat him up very badly. The year slipped, IBM didn't meet its commitments. That's the story.

Frana: Now heads rolled over this other than just Brian getting sent to Siberia.

Henry: Well…

Frana: Ben Persons is still wringing his hands over having to fire so many managers at that point.

Henry: This is the point where Roger Taylor disappears. My head didn't roll. I was needed at that time. But what we did is Hal Martin took the reins and we had a weekly review and we met with him every week. We would meet on Thursdays, because I would be up all night long Wednesday night making hand-drawn charts, because of my weekly review. You don't have time – there was a status sheet due every week. And what I did was assign one of my good guys, Jim Flynn, who is unfortunately deceased, but he was my best manager, a guy I really trusted, right, a good guy. I assigned him the job, and in addition to my report he would report. And Hal Martin told him he'd better be reporting the truth, 'Don't mind what Glenn says.' So we had to do this, this was an important sort of concept. You had two people. He was working for me, but he was also reporting right, on his status. And he was one of the critical pieces, but he was my main man and he was really good. Schleicher worked for him at the time.

Frana: Okay.

Henry: That’s where you know Schleicher and hearing about Jim Flynn. And so we pushed ahead a week at a time and we shipped the following summer, actually shipped, I think, in June.

Frana: That’s of 1980?

Henry: 1980. And I left Rochester immediately for two reasons. Along the way, my wife who had never got a good job, and hated Rochester because of the weather – she was from Florida – reminded me that this was a two year assignment. What’s going on here? And we had our son and she’s still working and this child and I’m working all the time, so it was one of these, ‘I’m moving you can come or not come’ things. But that was the first reason. And the other reason was, after going through all this I needed to get out. No one ever said, ‘Glenn why don’t you leave?’ But that’s appropriate. I mean they treated me very well, I went off and got another good job. They offered to move me wherever I wanted to. I was never mistreated by IBM at all. But, they didn’t have to say anything. It was appropriate. I’d been full-time managing the project for seven years, right and driving the project for nine years, it was appropriate that I get out of the way and leave. So after that, I walked away and I have never even been back to Rochester.

Frana: You’ve never been back to Rochester?

Henry: I keep meaning to go. And now my son wants to go because you know he was born in '73 there. And…that's not true, I went back for one quick business trip, didn't see anyone, and was out. I've been back one time. I still have fond memories of the people that worked for me; they worked hard and were good people. Even Roger, who was treated harshly partially and partially not, but this was a very strenuous kind of thing. And I was the boss and I had to basically make decisions who to get rid of or not. I had to make the hard decisions. There were many controversial decisions. So it was that part of my life is done, I'm moving on. This is the thing, you know, one of my classic interview questions when I interview people here is, 'What have you done that you are most proud of?' The thing I've done that I am most proud of is the System 38 and I've done lots of things since then, but this is a very innovative system done by a bunch of young kids in the cornfields of Rochester, which was a backwater of IBM, and the system's key concepts are modern today, the system has survived today. I am really proud of the team. I'm proud of what I did there. Yes, so it took seven years to develop it, I mean, but it represented at the time, in fact at one point I made a chart, you know typical ACM paper, the ten great advances of computer science – I made a list – we had seven of those on our system. You know, tags, storage, single level storage, integrated database, you know those types of things. We had built these into a commercial system and made it work and delivered it to customers, right, with little help from anybody. Research didn't help us at all. In fact, I have a very bad relationship with IBM research, and it started here. They didn't support us. You would think that they would have but they didn't. The forces of the mainframe people were constantly against us, constantly chirping, trying to kill us because we were an annoyance to [them]. You know IBM was very competitive. They owned MVS and 360's and we were building this thing. They just saw it as some Skunk Works project that would perhaps interfere with their grand scheme of things.

Frana: But it went on when the hardware caught up to be the best selling small system in IBM history.

Henry: Yes, that's again – well, in fairness it went through a lot of evolution change, but the basis was still the System 38, so that's what I feel good about. Yes, we had over designed it for the hardware and we'd underestimated how difficult it was to build these new concepts. And we were young and naïve. And this was our second system, as the book says, but we got it done and that's fairly significant because IBM history is littered with projects that – for every project that makes it you know five die or something like this, right. And along the way there were many other projects that started, thousands of people, big deals, you know like FS. There were other minor versions that withered and died. IBM developing a new operating system doesn't happen very often. In fact, it hasn't developed a really new operating system in its mainframe era forever. It just keeps evolving the one they have.

Frana: Now, I know your time is valuable, but just to keep the chronology straight now, when you left Rochester you started working on the RT?

Henry: Not directly. By the way, I have as much time as you want because this is my history so, well, I say that within limit – we have until lunch, how is that?

Frana: Okay. That's wonderful.

Henry: Okay, fine. All right. Well that's another interesting story. I'll tell that story because that's the second thing I am most proud of that became a big deal. But I went to Austin just as a place to move. And they actually created a job for me. It wasn't a real job but it was you know a staff job, which IBM is big on. So I was the manager of planning for DPWP, whatever. We are reporting to the lab director. It was sort of a made up job, staff job. And immediately, IBM decided upon a giant reorganization, IBM-wide reorganization. So I was assigned to a task force in New York and literally spent weeks up there, commuting on the weekend. And the task force reorganized and turned the division at Austin into a sort of a word processing, office-oriented division, OPD, something like that, I don't remember the name. And in that they had a major group, they had a huge multi-hundred-person project that was going to do a follow on to a thing called the IBM display writer. It's hard to believe these days but they had dedicated machines to do word processing; now we use pc's. And I was the technical guy again so they said, 'Okay, you become manager of advanced technology.' And what that meant was that I get all these little projects that were lying around with nothing to do, fine go and play with them and make something out of them. Well, one of the projects I gathered up was a group that was building a RISC microprocessor. The concept had been developed elsewhere but none had been built commercially. I mean it was a university concept. This was 1981. And so I said, 'Okay. I've got this processor. What can I do with it?' So I went back and I said, 'I like building systems for individual users.' And at that time, the IBM PC project was just forming or trying to form in Boca Raton. So I said they are doing that, I said what we should be doing is a more powerful version of that. A single-user thing. So I had a RISC processor, and I had no software and I had no applications. So I said, ‘Well, I don’t want to develop a new operating system. I’ve been through that. It’s too hard. People won’t accept it. Let’s go get this brand new thing called UNIX and get it put into our machine.’ So I was the first manager at IBM to speak of UNIX and my group did the first UNIX project. Later I became the IBM-wide manager of UNIX strategy.

Frana: This was Berkeley’s UNIX or Bell Labs’ version?

Henry: It was Bell Labs’ version imported by a company whose name I can’t remember well. Yes, it was the Bell Labs’ version. Just because there was a company to commercially support it. So no one knew anything about UNIX including me, other than just reading papers and things. Then I said, ‘Well we need some framework of this thing, because no one will know what we are talking about.’ So I did a deal with Carnegie Mellon University to build a workstation as a joint development project. They would do software and we would do a UNIX – well, it’s now easily called a UNIX workstation, we didn’t know what to call it in those days.

Frana: And where were you located at the time?

Henry: Here in Austin.

Frana: You were here in Austin?

Henry: Yes, right. So I have twenty guys building a processor and I collected another twenty guys, you know. And I'm making charts of a system, right. And a couple of software guys were doing this. But I needed more money, and I went to my bosses and said, I need more money and they all wanted no part of this thing. There was this main street project down the hall, four hundred people, building a display writer follow on. You know I'm over here with forty guys building something no one ever heard of, a UNIX processor. So literally, line after line of my management would say, 'Well, why don't you go to the next guy,' until I got to the president of our division who wouldn't touch it. He said, 'Why don't you go to my boss?' Well his boss happened to be John Akers, who later became the CEO of IBM. At that time, he was a group executive. So I went to New York again with my charts to John Akers. And there's a whole room full of executives, every one of whom had refused my request for help, except they are all waiting to see me die, right. It was me presenting to John Akers. I said, 'Here's this new kind of thing, it's got this UNIX thing which the universities like. It's got this RISC workstation, the concept started in universities. You know it looks like this. It can do these kinds of things and we'll do it together with Carnegie Mellon.' And John said, 'What do you want to do it?' And I put up the chart, and I said, 'I'll need so much and I'll do it by then.' The guy was across from me. He reached over and took my chart, folded it up and put it in his pocket and said, 'I'm agreeing with this, I'm going to remember what you said in this chart.' And off he went to become president. And there never was any formality with it. What I did was go around and say, ‘Well, I went to John Akers and he said to do it.’

Frana: [laughter] Put the charts in his pocket.

Henry: Remembered my words. Got this, put it together and we were building engineering scientific workstations, you know unit of 50, it was a technology effort. Go get UNIX on it, test it with university guys. And all of the sudden one day, I get called in and they say, ‘Glenn, we realize the error of our ways. This display writer crap is no good, it is never going to make it, and we’re killing that project. We are going to make your project the mainstream one and by the way, here’s four hundred people.’ This is like the kiss of death. It really cost us two years on a schedule, screwed up our competitiveness something bad. Because what they were building was a commercial thing, we were building an engineering scientific workstation. Once again, my engineering scientific thing was ordered to be a commercial thing. And we had a UNIX operating system, which is not the worlds’ easiest operating system to use. Not now, certainly not then, but it was fine for university kids. It had nothing to do with this thing. And so we just fought back and forth. We had to take people off the floating-point engine we were building because who needs floating point? It was a giant classic IBM disaster. In fact, it has been referenced in a couple of books just briefly. There were a number of books on this that briefly mentioned this story. It’s a classic story recognizing this project is no good, finding something else that’s alive, burdening that with this. And of course the four hundred people I didn’t need. It took me years to get down to the team I needed. But that thing became the RT PC.

Frana: Okay.

Henry: I’m proud of starting it, getting it done, through this horrible IBM thing. But it wasn’t a good product in the end because it had been delayed two years and it had been warped. It was half scientific workstation and half a display writer follow on – which is, neither. But, we got it done, and it was the first IBM RISC – it was the first product that had a RISC processor in it. And along the way, people had become interested in UNIX and for a while I had to manage UNIX.

Tape 2: side A

Henry: The RISC processor is different, but the software is AIX [Advanced Interactive eXecutive], which is our version of UNIX, the IBM thing. We were taking UNIX and we enhanced it a lot, right. That was called AIX. So we did the first RISC processor UNIX workstation, which was the forerunner very directly of the Power PC line, which is now quite popular. So again, my claim that my two major projects, the System 38 and the RT PC were the forerunners of AS 400 and Power PC, which are the two most viable things that IBM have.

Frana: That’s quite impressive.

Henry: That’s why I was made an IBM Fellow. They probably also made me an IBM Fellow to keep me quiet because I always had an opinion about what was right or wrong. But I wouldn’t keep quiet.

Frana: I thought they wanted those opinions. I thought that part of your duty was to prepare white papers and tell IBM what they needed to do.

Henry: Well, maybe they wanted it, but they didn’t agree with me. What got me this job, in fact, is an IBM Fellow project I started so…can I briefly tell you that?

Frana: Yes, sure. Go right ahead. We’ve got time.

Henry: So, this is 1985…

Frana: I wanted to ask you where you got the C language programmers? Were these Carnegie Mellon kids or I suppose you all had to learn?

Henry: We just learned C. I remember getting C on – and naturally I got one of the first IBM PCs and I remember getting a stack of the schedules of C compilers from someone. We just learned C.

Frana: Were the kids learning it in school?

Henry: I think so. UNIX was pretty much taking over by that point. But it wasn't as big of a deal as it is today. Learning C for a programmer is no big deal. C is a good language. I love it. We all had learned horrible things like PL/I, which I think is a horrible language – much more complex, much more baroque, et cetera. So learning C is not a big deal, especially because it is a good language. People liked it.

Frana: And to go back just a little bit even further…

Henry: Yes.

Frana: When did you start seeing well-honed computer science majors being recruited? Is that mid-'70's?

Henry: Mid '70's. Yes, sometime. Probably I would say after the 70's. Not that they were of great value to us, I mean, other than learning basic programming things. During the '70's there was this giant – you know we also went through this whole phase of top down structured programming, Harlan Mills, the whole 'how to do programming.'

Frana: Structured Programming. Structured analysis.

Henry: Structured, formal verification, all kinds of things like this and those were all well and good, but the reality is all we needed people to do was to be able to know how to be programmers and learn basic operating system concepts. I'll go back to the System 38. What we were building was so different than anything they'd seen in school, right. And a good programmer – there were great programmers in my day long before structured programming. I believe there is still a great deal of art in programming. Anyway, in 1985, when I became an IBM Fellow, I said, 'Hey, the hot thing of the future is going to be pc's.' IBM had just introduced the PC AT, which was really a breakthrough of the second-generation PC. It was really great.

Frana: Now that's the follow on to the RT?

Henry: No, no. This is the real pc. The Intel…

Frana: Okay, this is the Intel pc.

Henry: Yes, see go back to that. While I was doing the RT PC, which wasn't pc's – this was the RISC space thing…

Frana: Right. The workstation.

Henry: I would have to argue, I would have to explain why it was more costly and not that much faster, than Don Estridge who I knew well, was doing with the thing in Boca, which was the IBM PC, you know the forerunner of the thing that sits on all of our desks. And the truth is, I became convinced that the pc, the Intel based pc, was the way to go. Because, I was constantly working myself closer to the people – in fact that's part of our big theme right here – is we are building processors for the common masses here. But it was clear to me that the IBM PC was going to have tremendously good cost. That we could never approach with a proprietary solution. It was true even in the mid-'80's, and now it is totally true. I mean the cost is just ludicrously low on all the components because there are so many people in competition right. The volumes are so high. So I said, 'I want to be a PC person.' And I actually transferred myself, I got myself transferred into the PC division, which was considered a scandal because, this was 1985, pc's were looked upon as a toy by everyone at IBM, including IBM research at the time. Maybe it would be a nice terminal to a mainframe – that was the mentality.

Frana: Right.

Henry: I saw them as an end-user device. I mean it goes back to 1970 when I built a giant pc, you know, the basic system.

Frana: This was even before pc sales were really starting to take off.

Henry: They are starting to take off but there are still a few hundred thousand a year. Maybe a million a year. It is not like our two hundred million a year now. And the software is still very primitive. They are all running DOS, right, you know. But all of the signs were there. All signs were there to some people including myself. I believe, if you can give power to individuals and make it easy and cost-effective they are going to want it, right. IBM executives did not believe that. They could not imagine. And I would go see my boss every month, who was actually in New York at this time. I actually worked for someone in New York. The man had never had a pc on his desk. He told me once that his daughter had got one and he had tried it and didn't know much about it. They had no concept of what a person would do with a computer. They didn't even have terminals of their own, they didn't use anything. They were IBM executives. But anyway, I started a project to build an Intel compatible processor inside of IBM. This was the most action-controversial thing I did, and that is what caused me to quit. This was 1985, and Intel didn't own the world like it does today. In fact, IBM invested quite a bit of money in Intel, to keep them from going bankrupt. They actually bought twenty percent of Intel in 1985, stock. Then later they sold it off for profit. And I said, 'Look, we've got all this technical talent – Burlington and Austin – we have thousands of people with RISC processors. The volume is going to be over here. Why don't we build one of these things?' Oh this is like standing up in church and saying there is no God. Because the church, at that time, inside of IBM, was proprietary. You invent it yourself. Remember things called the microchannel? Do it all yourself, don't use someone else's. But second of all, is that the Intel architecture is technically the work of the devil, which it is. It is extremely horrible. Once a semester I go down and teach a class, you know, just as a guest lecturer down at UT in one of their computer science classes on x86 architecture, and the students gasp with horror. How can anything be so baroque? So ugly? It's just not decent computer science. It just is what it is. So I was advocating we build this ugly thing, which was a copy of what Intel did, right, so it was just awful. The reason I proposed doing it is because sales were going to go from a million, to I didn't know what. But you know something a lot bigger.

Frana: You were going to reverse engineer the thing?

Henry: No. We were going to do it from scratch. Yes, reverse engineer only the instructions. But we were going to apply all the RISC techniques, which Intel wasn't doing at that time. So we knew how to build a better processor. At that time, IBM had much better processor design expertise than Intel did. I talked to the Intel guy. So I was made the coordinator technically. I was actually the IBM's official coordinator with Intel on architecture issues and stuff like that. But in the corner we were building from scratch our own thing. It became extremely politically controversial.

Frana: And this was part of your ability to pick your own project right?

Henry: Yes, but, you can't pick your own project and turn it into a big project. The truth is, I wanted to develop product. But this was not just a research project, I was going to do this. I could do it, right, as a real product. And it was counter strategic and it turned out that IBM's chairman at the time had gone to school with Andy Grove and just lots of different things. Anyway, then a couple of other things. The other thing was IBM's dealings with Microsoft on OS/2. Basically, IBM gave the operating system business to Microsoft and the processor business to Intel. This sounds easy to say it after, but they actually have a couple of books that reference me during those days. I fought against it, I saw it happening, and said we should do it differently, right. They didn’t want to listen so I just one day said I can’t take it anymore and I quit to work for my next door neighbor who was twenty-three years old at the time. It happened to be Michael Dell. At the time no one had ever heard of him, no one believed – and now Dell is a big deal – at the time it was a joke. My wife fainted, [laughter] my friends laughed, management. I had to go see the president of IBM to explain this deed. And of course, life has changed. Anyway, I am still back trying to build, what we are trying to do here is build a very low-cost Intel processor to facilitate low cost pc’s to be used by the masses. You know the masses are not in the United States. We’ve all got three pc’s, and got plenty of disposable income. The masses are in China, India, Indonesia and Eastern Europe. But they don’t have a lot of money. I don’t mean to give you a sales pitch.

Frana: No, that’s okay. I saw some of the copy on your website. Yes, I think it’s important given that half of the world still hasn’t made a phone call.

Henry: The phone is an interesting analogy. Phones, televisions, do things that ordinary people want, that’s been proven. Pc’s can do things that ordinary people want to do, particularly with the Internet. That’s an application. It’s like a big phone and a big television – it’s a better phone, a better television. So the killer application is there if we can lower the price of pc’s. So our parent company is actually, because it takes a lot of money to do this, IBM kind of funding. Our parent company is a Taiwanese company who understands this. Taiwan builds low cost things and sells them into that part of the world. So I've worked my way down.

Frana: But your System 32, 38 experiences are very good for this is because it is all about price points and you are what sub 1000?

Henry: Yes, well actually they are good for many senses. But, yes. I always worked at every point in time at the lowest cost thing I could at IBM, the lowest priced thing, to go back. I worked on the 1130. It's follow on the System 3 Model 6, which is lower System 3. Then the System 32, et cetera, in our time our workstation was the lowest cost workstation. By the way, the RT PC, if it had come out two years ahead of time when it could have, it would have beat Sun to the marketplace. And it maybe would have been a different issue and it would have the right hardware and software on there. You know, we traded our schedule and we threw away our advantage for a fancy display, et cetera. At the same time, Sun started, you know, in 1982-83, right…

Frana: Yes.

Henry: And came up with the same concept and shipped in '86. So we were right there, with a similar concept at same time, burdened by IBM, instead of being a startup. Anyway, yes, so I've always worked on, to me, the lowest price thing. The thing, I've always tried to work on things as close to usable by people as possible. As opposed to – this gets back to my original experience where it was fun to be a high priest, right, but that wasn't going to grow the marketplace any – the days of computing as I've grew up with them.

Frana: It would be nice to talk to you another time about your time at Dell. But did you go back to the Bay Area then for a while?

Henry: No, I stayed here in Austin.

Frana: You stayed here.

Henry: My son went off to Stanford while I was working at Dell, so I spent a lot of time in the Bay Area. I still have roots there. Austin happens to be a convenient place to live, hard to get out of.

Frana: Now one final question for you that occurred to me looking through some of your promotional copy for potential employees, and that is that you've dispensed with a lot of the management bureaucracy. Everyone reports directly to you if they have problems. Is that a reaction to the layers and layers of bureaucracy that you found at IBM?

Henry: Yes, that was the second value, if you will. IBM taught me a lot of good things and a lot of things not to do too. So yes when I started the company I had two goals in mind. One was the product, building this low cost pc. The other was I wanted to do it a certain way. When I went from IBM to Dell, I saw an awful lot of this. Dell, at the time, was doing with one person what IBM did with a whole department, right. I didn't understand that when I went there, but I very soon did. Subsequently, Dell has become big and has got some of the same problems. So I came here and I said we are going to do it a certain way. A way that I had concluded was the proper way to do things. But it was obviously based on my experience. That's the only way you can conclude things. And so, there are still a lot of things that I learned at IBM that are still quite useful. IBM does teach you how to manage in many different ways. But probably more than what IBM taught me, [I learned from] the experience of going through these projects. I had six, seven hundred people, on System 38. That's how you learn not to do projects that way. I remember when you are a young manager, most managers are the same way, they think that hiring more people, having more people working for them is some sign of power, right, some sign of importance. It is to some degree at IBM, but IBM just assumes that if you want to do a big project you have to have lots of people to do a big project. Of course, your ability to do anything actually goes down if you have more people. It's my version of *The Mythical Man-Month.* By the way, I keep thinking that I want to write a book about what I've experienced here. Because what we're doing here with about twenty-five designers, the whole company is seventy, but the core design team is less than thirty, is designing a Pentium III compatible processor which takes thousands of people at Intel, and hundreds at AMD [Advanced Micro Devices]. Those are the only two companies we compete with and we are doing it faster and with far fewer people. So, yes, there's a whole very specific culture with how to do things with a small number of people. Basically, our people are really good people, really experienced people, and I do the things it takes to keep them. I get out of their way, no management, no bull, no meetings, no status reports, no system assurance, no checks and balances. Hire good people and trust them. Develop really good tools and methodology. IBM wasn't always crazy about that. I remember an engineer would cost IBM three hundred thousand dollars, you know, the classic accounting thing. And they'd gripe and moan, you know, about giving them pc's and things like this. We have tons of capital equipment here. We have few people, but our capital equipment per person is high. You know tools and things.

Frana: So IBM was capital poor with lots of human resources?

Henry: Yes, in my opinion. One of the controversial things is that there are stars. Hardware and certainly in software, there really are stars. There are people that can do the work of five other people. IBM would not recognize them and treat them importantly. At IBM, when I was a manager at least, every department had usually a really good guy and he would get an eight percent a year raise. And every department would have a guy who slept all day, woke up at noon and read the *Wall Street Journal* and he would get a four percent raise. You know, they wouldn't aggressively get rid of the sleeper, and give the star a twenty percent raise, which is what should happen. So we work on the star system. Our salary plan is very skewed. So it's a whole very conscious strategy of how to get good people, keep them, keep them motivated, and make them extremely productive. But at the heart is get good people, trust them and get out of their way. So yes, I manage fifty five. It's not true that we don't have managers, we've got a couple of support groups [that] have managers. But the development group works for me. I have fifty-five people working for me and I do technical work on the project myself and it all works fine. Where at IBM, if you have more than ten people working for you it's considered to be that you are out of balance.

Frana: So do you hand out copies of *Small is Beautiful*, as well?

Henry: No, but that's – we are the 'Small is Beautiful' reference. But it wouldn't work at IBM because, I mean this is a problem that Estridge ran into, you know the story about Estridge and the PC. This is one of the great stories. He was able to build the first PC, it was a small isolated group that no one thought was important, that no one thought would come to anything. As soon as it became something, it became a division, dragged Estridge down when it became…couldn't do anything. Hundreds of people, thousands of people, right, that's a horror story in itself.

Frana: Are there other companies that you perhaps not emulate yourself after, but are doing it this way?

Henry: Well, I think a lot of the concepts that I think any startup uses, you know five guys in an office, right, and starting a company will follow the same general things. I don't know if they've got them expostulated as clearly or not. And I don't know if they stick with them. Most small companies by the time they get our size have hired a professional manager, have created – the first thing that a lot of startups do in town, they start flat almost by definition. The first thing they do is give themselves titles and then they give titles to new guys coming in. We have no titles here other than me. I am the president, everyone else is a senior engineer. I mean you start to have vice-presidents of engineering, you have a hierarchy that is going to slow productivity and slow decision-making. So I don't think anyone has carried the concept of a perfectly flat, no hierarchy, we don't even have team leaders. Just a bunch of guys working together as a team doing something, with one guy working for one guy, as far as we've carried it. People start that way but they immediately turn into – one of things that helps is that we have no ambition here. Actually we have a lot of ambition, our ambition is to have our products be successful and be on the low-end processors. But to do that we don't need huge amounts of people here. The sales and support, the manufacturing is done in Taiwan, right. We are designers. Most people that start a company have dreams of going public, making a lot of money, hiring a lot of people, and becoming famous. One of the things that is interesting at this point is that, I'd always like to have more money, but basically, this is my retirement job, I'm going nowhere. I don't even care that much. I want more money, but I'm not doing it for money. I'm not going anywhere else, I'm not looking to pad my resume, which actually makes it incredibly easy to focus on what matters and not worrying about looking good, looking big, having good titles, things like that.

Frana: You spent too much time doing that at other points in your career?

Henry: Well, that's again one of the things we talked about the advantage of doing things alone. You know, at Dell I managed over a thousand people, and as a senior vice-president I worked directly for Michael for a long period of time. Having done those things, I've been there, done those things, and they are not interesting for me. Therefore, I can focus on what needs to get done. Most managers that are doing a job, are heading somewhere else. They are doing that job as a stepping-stone to some other job, which warps their job. I'm doing this as my last job.

Frana: Well, I thank you for your time Glenn, and I do hope that you do a book project. And I'll send you a copy of this and maybe this will stimulate that process.

Henry: Well, I enjoyed it.

END OF INTERVIEW