

HISTORY OF MS-DOS
INTERVIEW WITH BILL GATES

Plaintiff's Exhibit

7429

Gomes V. Microsoft

MSC 00567023

CONFIDENTIAL

RBC 001029

Transcriber's Notes: All questions and comments from the interviewers are denoted by a "Q." entry preceding the question or comment (there are very few). All other paragraphs are transcriptions of Bill Gate's speech. Check all names of people, products and companies--I don't guarantee accuracy of spelling here. (Pam Beason, transcriber)

(Preliminary conversation, introductions, etc.)

When microcomputers started out, we didn't have operating systems--we just had BASIC built onto the machines. We actually did a version of BASIC called Disk BASIC (?) that had all the file management stuff embedded into BASIC--we did that on the very first floppy disks that existed for micros, Altair. That was back when floppy disks were very "flakey", but it was quite a step up from, at the low end, using paper tape or cassette or, at the high end, having to buy very, very expensive hard disks. So Disk BASIC back on the Altair was very, very popular.

In parallel with our Disk BASIC, Gary Kildall (?) at Digital Research had written the Control Program for Microcomputers...I never figured out why the slash was where it was, but... We at MITS...Microsoft was very closely related to MITS back then: I was never an employee of MITS who did the Altair computer, but Paul was--he was the Vice President of Software Development. I was Microsoft, in fact, for quite a while I was the only full-time employee at Microsoft because Paul was the only other person and he was working there at MITS.

We went out and looked at CP/M 80 and we knew it was pretty simple to write a file system, so actually MITS for a long time had their own operating system -- they took the code out of our MITS BASIC and built their own operating system. We never pushed that with other manufacturers. People wanted to do more than write BASIC programs --they wanted to run other languages, they wanted an operating system that was broken out, and a lot of the hobbyists went on an individual level and bought Gary's thing. And then this company Lifeboat Associates came along ...maybe this is all too historical ... Lifeboat Associates came along and started actually offering stuff in CP/M 80 format. It was probably Lifeboat more than anybody who really got things going. They took our Fortran and our COBOL and a bunch of public domain software and put it out for CP/M-80 and then people like Northstar and Prosser Tech? decided that it would be nice to have CP/M-80 and then finally Insight ? which had their own operating system decided to have CP/M-80 and so the thing got a following to the degree that most of the new machines put it out.

Macropolis (?) had their own operating system, and they trashed that...there was such a tower of Babel in operating systems. Cromemco did an imitation of CP/M-80, called C-DOS ?, but there was compatible- CP/M-80, and Technical Design Labs did their own imitation of CP/M-80, which used the Z-80 very effectively. It

CONFIDENTIAL MSC 00567024

RBC 001030

was a very, very simple operating system, and the main benefit it had was that you could buy any variety of disk, whether it be big hard disks, or different types of floppy disks, and it would virtualize the interface for the file I/O. As far as screen I/O went, the basic philosophy back then was that you did either just teletype type I/O through the OS, or if you wanted to do screen I/O, you'd have an installation process, where the application would give you a menu of things to choose from, and you'd say what your screen was, and then it would use the right escape sequence or direct memory commands to do that. So things like WordStar had this install process, and things like DBASE had the install process, and things like our languages just used the teletype I/O--that was all fine.

So, actually, there got to be a fair amount of CP/M-80 applications, we did the SoftCard, which was Paul Allen's idea of sticking a Z-80 in a 6502. Our languages on CP/M-80 were the most popular thing because of the nature of computer users back then. Wordstar also was incredibly popular, amazingly, as time went on, and DBASE came out--well, actually DBASE was a little less clear that it was the leader...there was Commodore and a bunch of stuff...

So everybody got the idea that you didn't want just BASIC, that you really wanted a operating system to virtualize things. And CP/M was pretty stagnant for a long time. We decided to write our own 8-bit operating system--which we did...we wrote this thing called M-DOS because we wanted to have real multitasking, good performance and a better disk structure and a lot of stuff like that... but by the time we got it done it was fairly big, and of course 8-bit processors couldn't hold that much...

Q. Was it actually a multitasking operating system?

Oh, yeah, it was was multitasking 8-bit operating system, written by a guy named Mark MacDonald, ... one of the guys who left Microsoft and later joined Paul to work in the group up there actually was the author of M-DOS. And the thing was a little harder to write than we actually thought it would be. By the time we got M-DOS done, we only sold it to a couple of customers in Japan, because we had gotten so many people to buy CP/M-80 by then that it was fairly entrenched.

We thought, OK, 16-bit processors are a big opportunity to do an operating system, because it was fairly clear over time you didn't want to have BASIC in the middle to virtualize all that disk I/O--you really did want an operating system. About this time we took a license out for Unix, and we did Unix for the PDP-11 and Unix for the Z-8000, and Gordon Letwin, who is now the chief architect of MS-DOS, was off in that Unix area for a long time.

We ported it to various ^{versions} microprocessors -- that, we thought, was nice, but we had (a vision?) that the restrictions and royalties from Bell Laboratories made...we weren't going to make that

MSC 00567025

CONFIDENTIAL

RBC 001031

totally a mainstream thing that was going to be on millions of machines. Our vision was more single-user network than multiuser, so although XENIX served for our own development needs and for people who wanted to use micros to replace minis, we knew that we wanted to do our own operating system, that was more lean, and more appropriate for just what somebody wanted in an office automation work station.

Q. Was this before IBM?

Oh, yeah, this was way before IBM. We were doing XENIX stuff two years before IBM came along.

Q. And you were thinking 16-bit operating systems at that point?

Well, I mean M-DOS was sort of a ... See, in 64K, you've got to cram everything in 64K, the O.S., and the application and the screen and everything, and M-DOS had come out at like 20k...it was nicely done, lots of it. But we had definitely decided that we needed a new foundation before we could get to critical mass, and XENIX was starting to sell well.

Anyway, then the 8086 came along. We decided that we would really jump on the 8086, and in fact I had said to Paul that I wanted to pioneer the 16-bit stuff quite early, because that's when I told him that I didn't want to develop anything else for 8-bit computers. What we had done was, for the 8080/2-80, we had done everything--linker, assembler, COBOL, Fortran, all the stuff, and then we had done BASIC for all 8-bit processors, 6800, 68502, actually 9900, which you wouldn't even call an 8-bit processor...we had done BASIC for a ton of stuff, and the question was, whether to fill out the grid and have Fortran, say for the 68502, which was the next most popular 8-bit processor, and I said no, let's go to 16-bit, and that's really ...It was only a couple of weeks later that he invented the idea of the SoftCard, because that was the only way that we would sell Fortran to Apple users, and of course that later became the most popular form of CP/M-80, we went got a flat fee from that, became a customer of Gary's for \$46,000, fixed fee for all time.

We had done a lot of CP/M-80 adaptations, because they really weren't into doing adaptations and helping customers out to do things, and we were (porting?) a fantastic amount of business to them.

Q. What was about the date when you made the decision that you didn't want to do any more development for 8-bit?

Ok, when did we decide that? ... That was early '79. We had really talked about it in '78, because we were talking about our move into applications... and when we would do that... you've got to understand that in '78, it was a company of 14 people. We moved to Seattle on January 1, 1979, the company had 14 people, and ... during '79 we focused on the 8086, and that would've been

MSC 00567026

CONFIDENTIAL

RBC 001032

probably because we've got to integrate...

It's really complicated the way you have a ROM BASIC, and if you don't have a disk that boots up in ROM BASIC, but if you have the Disk BASIC, the Disk BASIC comes on top of that and hooks into the ROM. It's very, very complex, because even though the 8086 has one megabyte, the magical 64K is still a very important number, because although it's not binary-compatible, they routine really a lot of the 8080 architecture, so 64K is still a very important part.

So the whole way the BASIC worked with the ROM and sometimes have disk and sometimes not was pretty messy. There had to be a close coupling in figuring out who was going to write the I/O routines, and who was going to test them, and um... Oh, meanwhile we were trying to convince IBM to really go with the 16-bit processor, they were still thinking it might be 8-bit. But a couple of people on the design team also wanted to go 16-bit, and we were saying how great it would be -- it kind of violated the rules of designing the machine around existing software, because as far as 16-bit goes, the only thing we had running at that time was BASIC, we had some "craw (?)" software working, but that's not important...craw software means that the compiler runs on an 8080 and it generates 8086 code, which is kind of a strange thing to have, but we had it.

Q. Did IBM look at going 16-bit as more of a risk because there wasn't all this software?

Right. We were going to have, well, hardware side is always easy, but we were going to have to write all this new software. We were committing to write Fortran-86, which wasn't done, Pascal-86 which wasn't done, assembler-86, loader-86, all this stuff which wasn't done. The amount of code we committed to the first day was like 400K of code. So then they started saying, yeah, they thought the 16-bit thing would be good...we would do a consulting contract where you write the design up, so we showed them how we wanted to do graphics, we showed them how to do keyboard and stuff, and there were a lot of smart guys in the IBM side. We had a lot more experience in personal computers. They were very receptive. So the 16-bit thing got in good shape, but then they couldn't get what they wanted from Digital Research in terms of time, dates and .. by then we changed our mind about the wiseness of involving Digital Research in terms of that being a strategic thing, and would Digital Research perform, and how they had done on some previous things, and Kay Nishi had said one night, look, let's just do it! It's kind of a famous night conversation where ... it was just Paul, and Kay and I, and Kay gets up and says "Let's do it. Let's do it."

Because we had heard rumors -- I don't know how much Paul talked about this, but we had heard rumors that Digital Research was buying compiler Systems (rogue?) C-BASIC, our worst competitor in the BASIC area, and that whenever somebody came to get CP/M, they would be bundling those together, something that would make it

CONFIDENTIAL

MSC 00567030

RBC 001033

converted it over to MASM, which is the more structure type assembler that uses typing in the way that Intel decided they would for the 8086. So we did DOS 1, and we were in a very serious competitive situation, with MS-DOS vs CP/M in those days. I mean, very serious. When it first came out, Future Computing called it the "CP/M Record Player", and said oh, this is a CP/M machine; and in fact CP/M was announced as a thing... very late in the game through a series of manuevers, Digital Research got IBM to agree to offer their operating system as well, it was very, very late in the game... but they did sort of say, hey, this is an important machine, and maybe we ought to be on it, and they insisted on doing packaged products and charging a lot of money, so it ended up being priced like \$300, and we were priced, at what was it, \$40; we'd gone 40, we're now \$65, I forget what the price was, maybe it's 50, anyway, they priced DOS fairly inexpensively ... the machine was announced September 12, 1981, and they actually shipped it in October 1981, and by then we'd all realized that what Microsoft had been saying all along was true, which was that the disk machine was the important machine, and that the non-disk machine was out of (?) so virtually every machine sold MS-DOS.

In fact, it more than outsold the machine, because the only way to get the manual was to buy MS-DOS, it was \$50 bucks or something...you should get back and look this up, because I've forgotten exactly,... and so a lot of MS-DOS and we were real happy that they were buying that, and we were promoting people to write MS-DOS-based software, but for the first year and a half the machine was out, if you ever look in the magazines about software you'll see it says MS-DOS and CP/M, and some say only CP/M... and so, then there's this whole name confusion...I don't know if anyone has been able to figure out when we named it, but when we sold it to IBM, it was not called MS-DOS. There's so many names for the software in each system, um, Tim called it SCP DOS, for a while we called it 86-DOS, I don't know why we switched to that, I guess we switched because we didn't like that, ... then IBM did not, absolutely did not call it PC-DOS when they first offered the machine, it was the Personal Computer Disk Operating System, it was never called PC-DOS. It was a rewrite of history to say that they called it PC-DOS. It was just called, we called it, by the time they introduced it we decided to call it MS-DOS, but it was in between when we licensed it and when we introduced it that we decided to call it that.

Meanwhile, I had Eddie Cray (?) at Lifeboat agree to help me promote MS-DOS and help us get behind getting a lot of applications for it. But they decided to call the thing Software Boss 86, because they had a CP/M clone called SB 80, Software Boss 80, and they wanted to have this family thing called Software Boss 86, and they owned the trademark for Software Boss, so I couldn't use SB 86, and we said MS-DOS. Our first new licensees all decided, you know, Compaq on the first version called it Compaq DOS, Xenith called it Xenith-DOS, then I said, look, this is bad news... we've got to get everybody to call this thing MS-DOS. And today, I've got everybody except IBM calling

MSC 00567033

CONFIDENTIAL

RBC 001034

it MS-DOS, Compaq, Xenith, and all these people... And MS-DOS is a trademark, and so it's all cleaned up.

Q. Who called it Q-DOS, Quick and Dirty DOS?

I don't remember it ever being called Q-DOS, maybe when we first brought it up on the PC we called it that for a while...but that name never saw the press. This name was absolutely in all the Lifeboat advertising for the thing, then of course PC-DOS came up, and now people act like IBM called it PC-DOS from the beginning. In fact, an IBM guy, a communications guy wrote a letter saying that they called it PC-DOS in the beginning, and that's just not the case.

So DOS 1 got out there, and we wanted to get a lot of people to write applications for it. IBM wasn't shipping in Europe, and CP/M was relatively more entrenched in Europe, and Digital Research did a clever thing, they got some people to do machines that combined both 8080 machines and 8086s together, like the DEC Rainbow, it's not well remembered today that that was a dual processor machine. And in fact, well, that's kind of clever in a way, because it means that you want, it makes it more important to have CP/M compatibility when you're mixing 8-bit software and 16-bit software on the machine. We didn't use the same disk structure that CP/M-80 did, our file format was different. They used kind of an unusual FCB thing--we supported FCBs from an application interface point of view, but the way we laid stuff out on the disk, it's totally different. So that meant that people had to write conversion programs, when people did these little Z-80 plug-in cards for IBM machines, some of them used CP/M-80 physical format, logical format, and some of them used MS-DOS logical format, and just emulated CP/M-80 on top of that. The Baby Blue (?) card did it right, they used MS-DOS format on the disk and emulated it. IBM absolutely was not enough to create a standard for MS-DOS. We had to go to, well, Victor was a very key company, it's not well remembered since their sales have declined since they went to Chapter 11, but Chuck Tuttle was very successful over in Europe, initially he had done some CP/M-86 stuff, and so we licensed him MS-DOS and by doing a lot of special support work for Chuck, we got him to offer more and more things, like the networking and the languages only on MS-DOS, and we got things pushing toward MS-DOS. Over in Japan, Mitsubishi got involved in CP/M-86, and we did Multiplan and Fortran for them on top of CP/M-86 and then Dec, here in the United States, on the Rainbow, was pushing CP/M-86, they didn't have any MS-DOS at all. After a while we got them to offer both MS-DOS and CP/M-86, but it was only about a year ago that we got them to offer most their stuff up on MS-DOS. It was was a real battle with Zenith, because they had an 8080 and an 8086, and there... it was really tough.

We ran an ads for MS-DOS, you won't find them any more, they're fairly classic, but just for fun you might try to find one and stick it in the book..., but they were classic ads..

Q. In BYTE?

CONFIDENTIAL

MSC 00567034

RBC 001035

Yeah, in BYTE, and COMPUTER SYSTEMS NEWS, and Electronic News, saying you ought to buy MS-DOS.

Q. Would this be in 1982?

Let's see...yeah. All in 1982. I think by 1983 we were starting to feel better. We kept trying to announce, we kept trying to tell the world, hey, here's a design win, we kept trying to count the number of applications programs, but all that Victor stuff in Europe and that Mitsubishi stuff in Japan was causing us real trouble, and Digital Research managed to act like CP/M really meant a compatible standard, and CP/M-80 was one set of programs and CP/M-86 was another. They had multitasking before we did, because we still don't have multitasking, and they did a lot of funny versions, like for the DisplayWriter. For a while they were selling those copies on the DisplayWriter, and that was going well, and there were these holdouts, like, who was it -- Gifford Computer? And there's somebody who still hasn't bought MS-DOS, I'm trying to think of who it is...it's this incredible holdout, it's not (Godbout?), Morrow (?), it's George Morrow. He's such a holdout. I think maybe he's licensed it now, in fact he license it about 6 months ago... he was such a holdout, and Godbout (?) was a holdout, and Gifford Computer was such a holdout, still doing things the CP/M way, and eventually there's this really tough decision that you have to make with your other products, which is, do you offer your other products on CP/M-86, or only on MS-DOS. Do you try to help your operating system, which sounds nice, or do you go down in smoke because you tie everything together?

Well, Digital Research in languages only offered their stuff on CP/M-86, and we offered it on both, and for a long time we offered our languages on both. We primarily offered our applications only on DOS, but Multiplan, in the case of Mitsubishi and DEC, we did on CP/M-86, because at first, we had a risk aversion, we weren't willing to put all our eggs in one basket, we sort of thought we were in good shape... anyway.

Over time, when you picked up the magazines, you saw less and less of CP/M-86. Today, I think if you did a poll of readers of PC WEEK, they probably couldn't tell you what CP/M is... and they certainly couldn't tell you, if you said "Was the IBM machine once viewed as a CP/M machine?" they would say "What? No way!" and in fact the only operating system besides DOS that even has a measureable market share today, as far as I know, is XENIX, because it fills a different set of needs, like that multiuser stuff, or people who want the UNIX compatibility.

In parallel with all this, we were doing DOS 2.0, and DOS 2.0 was, as far as IBM was concerned, they had a hard disk, and they wanted to support the hard disk, which meant the hierarchical file system. As far as we were concerned, it was our chance to put in independent device I/O and really do a lot of things we hadn't done in version 1 in terms of the debugger, you can type mnemonics (??) into the debugger, that's one of the things that

MSC 00567036

CONFIDENTIAL

RBC 001036

Chris Fetters did, so the DOS 2 spec was pretty open-ended. We were working on it, but as soon as they had the XTs ready, the hardware, then it was time to bring DOS 2 to completion. People were really surprised -- oh, DOS 2 is bigger than DOS 1, will you be able to copy it onto the disk with the application, people back then were thinking they would ship applications with DOS on the disk, which we were very opposed to. There was still some of that going on, then they knew they were going to do PC-Jr, so they wanted DOS 2 to be smaller, so we went through all these shrink things, and we made it smaller on the disk by finding the 9th sector, that's really why we went to the trouble of doing that ninth sector stuff, so there'd be more room on disks so that DOS would still fit on disks. We found out that in DOS 1 people had coded absolute addresses into their code, so they couldn't... you know, DOS 2 was bigger than DOS (and slower??) and we put that stuff in like the VisiCorp stuff, they wanted to relocate everything...except for VisiCalc, the other stuff wouldn't relocate, so it wouldn't run, and there's this problem in the initial IBM PC where it can't (DMA?) across a 64K boundary. People had bugs in their code where they didn't know about that, as as we pushed, as DOS got bigger, we moved their codes and 64K boundaries moved, and a few things, like Time Manager, ran into that.

We started to learn with DOS 2 how hard it is to upgrade the world's most popular operating system. Actually, it didn't become the world's most popular operating system until after DOS 2 was out there. And we told IBM, Ok, discontinue DOS 1, and they wanted to keep it around because maybe people needed it. We said memory's cheap, and it wasn't as cheap as it is now, so it took a little while for DOS 2 to catch on, but it was required for your XT, and so that helped a lot. Like a year ago, we got to the situation where nobody writes stuff for DOS 1 anymore, so they did finally discontinue the thing. We haven't sold anybody DOS 1 for so long...we had a few customers like Wang, who just became customers after DOS 1, they never had to mess around with DOS 1, they were DOS 2 to start out with. Compaq did do DOS 1, then they moved up to DOS 2... Victor did do DOS 1, then they moved up to DOS 2...

DOS 2 was where we did the good Kanji stuff, and we went back and attacked, attacked is not a good word, we offered good value to the Japanese market, and that enabled us to gain good market share, we really did the Kanji stuff right, and in fact they eventually had to just imitate our Kanji stuff in CP/M-86 because it was done so well it became a standard for how it was done. And we keep evolving a little bit to keep, you know, getting better and make it a moving situation.

But today, if you want to find CP/M-86 still, go to Japan. I mean, MS-DOS dominates, but there is measureable CP/M-86 still in Japan. The funny thing about Japan is, Standalone Disk BASIC really got out of control in Japan. Everybody wrote their applications on top of Standalone Disk BASIC, so up until a year ago Standalone Disk BASIC was the most popular operating system

MSC 00567036

CONFIDENTIAL

RBC 001037

because NEC has that on their machine.

And, you know in France, up until about a year ago there was thing called Prolog, which was this local operating system, that was very, very popular, but slowly but surely we beat it down. Every once in a while there'd be rumors like, oh, Seymour Rubenstein, he was this very competitive guy, he'd say he was only going to support CP/M-86, and in fact they got the Atlanta division to offer concurrent CP/M and some applications on top of it and then people thought that meant IBM was behind this, but in the end that didn't come to much...

Anyway, we got DOS 2 done, and that was a small team of people. I think DOS 2 was Aaron and Zibo,...

Q. Chris Peters?

Chris worked on the debugger, on a few of the utilities, he didn't work on that much stuff, he fixed some of the bugs and stuff, and there may have been one other guy besides those three, but really, in terms of full-time, it was really, was that all it was, I mean Tim had left by that time, he went back to SCP for a little while, and then he went back to Falcon, and then he also wrote another operating system, and the funny thing is, the circle came full term, we decided to do an 8-bit operating system that used the file format of MS-DOS, and that's this thing called MSX-DOS. It's not important here in the United States, there's no more Z-80 machines sold here in the United States. But in Japan, the most popular home computer, which we've sold millions of there, is MSX, which is a Z-80 based machine. And Digital Research wanted to sell CP/M-80 onto it, and of course we wanted to see our own, so after we finished MS-DOS, we eventually did ask Tim to write an 8-bit operating system that used the logical file format of MS-DOS, so he wrote for us MSX-DOS. That was when he was going to start Falcon, that was the seed money was the money we paid him to write MSX-DOS. So he banged the thing out, did a good job, and we turned it over to our guys in Japan to us. So that is actually a very popular operating system in Japan. You can take disks from MSX-DOS and stick them in MS-DOS, it's sort of like MS-DOS 1.0 in terms of its features. It's not a multitasking DOS, we learned that you don't want to put heavy-duty features in there.

MSC 00567037

CONFIDENTIAL

RBC 001038

Q. How about the hierarchical file structure? Is it only supported in ...

No, it only supports DOS. It's smart enough to, unlike if you take hierarchical disks back to DOS 1, it'll print out kind of these garbage things, because it didn't know to recognize hierarchical things, because when it was written we didn't have...MSX-DOS will say, hey there's hierarchies on that thing, I don't understand it, you've got to go flatten your file structure before I can read it. But it's flat. It does do the device driver stuff the MS-DOS way instead of the CP/M-80 way, but it's called compatible with CP/M-80, so you can run CP/M-80 software on top of it.

So anyway, ... so Tim was not involved and DOS 2 got done. We changed a lot of the utilities, and this pathname stuff... turned out to be a lot of work because we had to change a lot of the utilities, and somebody wanted a print spooler, and the thing wasn't really reentrant, so we sort of figured out how to do a print spooler under DOS 2, and there were a lot of challenges in DOS 2.

Q. How important was XENIX compatibility in your mind at that time?

XENIX? It was never very important. It was not important. We wanted to do an hierarchical file system. There's this random thing in DOS 2 where ... turns out in DOS 1 we used the slash key for switches and things, so you can't use the slash as a pathname separator, so, I can't remember whose idea it was now, they probably won't stick up and claim it now, but somebody decided to use backslash. I don't know if it was me or Larson, or who, but we decided to use backslash, which is different under any kind of Unix thing, but big deal, well then somebody got the idea to put in CONFIG.SYS this thing "Switchchar=", where you could change switchchar from slash to backslash. But then, I said, this is bad, because if people are distributing batch files, then how can you be compatible exchanging batch files when people have different switch characters. But then Wang's keyboard didn't have the backslash key available on it, and we never documented the switchchar feature, we never told people about it, but people always find out about everything, and use it in unusual ways. So it's taken us a while to wean people off of that switchchar feature.

MSC 00567038

There are a couple others, like /dev, anyway, there were some other unusual things. And there's always inherent stuff, like is CONTROL-Z an end of file mark. Well, that's all CP/M-80 stuff. They couldn't do exact file lengths in CP/M-80. We did exact file lengths in DOS, but then because of the way we were running CP/M-80 software on top of it, we started making certain things interpret CONTROL-Zs as logical end-of-files. But that's very bad news for binary files, and so that was a real mess for a long

CONFIDENTIAL

RBC 001039

time, as well. Then we got that straightened out. And then this dev thing was, do device names override filenames with the same name -- you know, that we weren't sure which way to go... and so we said hey, its just plain override, and we got rid of that /dev stuff. There were a lot of things that were judgment calls. You know DOS 2 had to process batch files the same as DOS 1, so if you had a thing like COPY A.* = *.B, you know that way that kind of thing worked in DOS 1 was kind of unusual, we had to mimic that unusualness in DOS 2. We wanted to have more error codes, but there weren't provisions for people to see more error codes. We wanted to expand Int 24, but people had their own Int 24 trappers. It's hard, I mean it's really hard to do a new release.

Then, after DOS 2... After DOS 2 came out, we got everybody to upgrade to DOS 2, nobody held out for DOS 1, we made it real easy for people to upgrade to DOS 2 financially, license-wise.

And then we wanted to do multitasking, and IBM wanted to do networking. And we decided to do networking. It's not as simple as that, I mean you can't just fill in the blanks, you can't just say we wanted to do X and they wanted to do Y and we decided to do Y, but in this case we weighed the factors, and we had other customers pushing the same direction, so we did it. And we knew that it was going to be kind of a mess to put a server on top of a non-multitasking operating system and pseudomultitask the thing, and it was, it was a sort of messy thing in the OS group, but we put the good international table stuff in and we fixed up a lot of stuff and DOS 3 was again, it was Aaron, and Zibo, and just a few other people helped out toward the end, people like Eric Evans, and so on, but not really that many, really DOS 3, it was really those four guys, Manny was in for a while, but certainly over 50% of the work that got done was those two guys. And the thing got very complicated; once again, the stuff with compatibility was very, very complex, and IBM had bugs in their net files hardware, and you've got to decide do you fix those in the software or the hardware, and you know there were performance issues, and we that ran old applications and passed them back new errors was very complicated, and FCB calls across the network were very complicated, and directory lookups First Search Next Search (??) the thing was complicated, but we wrote it, we got it done, it was a little bigger than we wanted it to be, we put in a patch area, we got it done.

And it wasn't until after the completion of DOS 3 that we got into the thing with the IBM PC, where IBM recognized the IBM PC to be the office automation work station and that's when they decided they wanted to really have a long-term arrangement with us and that's when we did that joint development agreement. And that's since then we've increased the amount of people architecting DOS, involved with DOS dramatically. Today it's more like 30, it's very close to 30, instead of the 3 who did the previous version. You see, people always used to say to us when we came out with DOS, anybody who was buying XENIX from us, do you still believe in XENIX? Well, you bet we do, we had more

MSC 00567039

CONFIDENTIAL

RBC 001040

people working on XENIX all the way up to the middle of DOS 3, we had more people working on XENIX than on DOS.

And even a super strong designer like Gordon wasn't shifted over to DOS only after that. In fact, a lot of the key DOS people now, like Anthony Short, Gordon Letwin, Adrian, a lot of them, are people who were initially a part of the XENIX effort and learned a lot. That's great in a way, because they learned about protected mode and multitasking and things like that.

XENIX-compatibility was never an important element. I mean, we're just not driving towards that. The large base is under MS-DOS, and the trick is to give new features to old applications, or give new features to new applications without interfering with old applications, and without adding to the complexity, and it's super hard.

Q. Well, from having the new file function calls, the pathname function calls in Version 2. They happen to be XENIX-compatible. Was that just because you used the same hierarchical file structure, or ...?

Well, they're not compatible in the sense you know can take the XENIX and stick it on top of the DOS, they're the same approach, and UNIX got that stuff from MULTIX, it's a very straightforward stream I/O approach, the FCB thing, the File Control Block thing, where the data structure describing the file is in the application's address space is very messy because then the application can destroy those things or move them around. You really want all the description of the currently active files to be up in the OS address space. Well, in these non-protected machines, sometimes you don't even make that distinction of what's OS address space and what's application address space, but we will as time goes on. That's why the handle thing is so much better. It's better with networking, where you want that information to be back up on the server, it's just cleaner than the FCB stuff was. The FCB stuff was causing us trouble and it's just a more complicated program to write to -- it's a stream I/O which allows you to merge in the pipe stuff better and device stuff better, was just a good decision, but then you had the case where we had to maintain the old system calls, as well, and we did.

MSC 00567040

CONFIDENTIAL

RBC 001041

Bill Gates
Side 2

Well, you'd have to get like Ballmer, or somebody like that to tell you the exact day. It was announced in August, 1985. It was signed in August, 1985. Actually, maybe it was signed in July.

Q. We could probably get Natalie to verify things like that, too, huh?

Yeah. We had discussed it with IBM starting in late '84.

Q. The future?

Well, you can't really say too much about the future. We always say we're going to use 286 protected mode, we're going to do multitasking, we believe in networking, we don't believe in multiuser, we believe in graphics user interface, but the way graphics user interface works, you don't just want to throw it into DOS one day all of a sudden and say everybody's gotta have this if they want the next new version of DOS. It's an evolutionary thing, where some people realize they want it, and some won't, so you know. Windows we've done as an extension of MS-DOS that's optional. We can improve the buffering and the networking and stuff like that and still Windows just sits on top of it. If Windows ever gets to the point where 90% of the people are using it, then maybe we'll just stick it together and call it one product. But... For now, we've decided to be flexible about that.

Q. Can you say just a little bit about how you see general industry changing and how that's likely to affect the future of MS-DOS?

Well, MS-DOS, fairly clear to us, is the office automation work station operating system, and when we say there'll be a machine on everybody's desk, we mean a machine running some future version of MS-DOS. And there's more software around today for MS-DOS than for any other thing. I used to give this old speech, you know, when I was trying to explain to people the phenomenal momentum behind MS-DOS, I'd show a list of 200 word processors, and then I'd show all the people who wrote letters in after all the magazines reviewed those 200, saying you forgot our word processor, and how could you do that. I mean the fact is, it creates this perfect competition between packages on top of one environment instead of everybody off in their own niche. It allows users to interchange stuff, and to a degree it's hard to learn, but you learn it once and that's it, and you can explain it to other people around you who are all dealing with the same thing. I mean, MS-DOS has got a life of its own that's very, very amazing, and primarily as an office automation operating system. It'll be used in other environments, but our priorities, as we've defined it and go forward, are for it to be used as an office automation work station and now we just talk to the

MSC 00567041

CONFIDENTIAL

RBC 001042

software developers. You better believe when we've got a new version we go see Lotus, and Ashton-Tate, and people like that to make sure they feel good about the stuff that's gonna be in there.

Q. Do they ever specifically make requests that influence the way that you solve your technical problems?

Oh, sure. I mean, there are very smart guys at those companies, and they have an influence on how we do stuff.

Q. The famous night that you said, let's do it, do you know the date of that?

It was in ...

Q. Where was it, was it in the office here?

Oh, yeah, it was up on the 8th floor in my, I had the corner office on the 8th floor in the, it's called the Old National Bank building, actually they've renamed it, now there's two buildings there, but it was in my office. I've changed my furniture too. Kay and I were just sitting there at night, and Paul was on the couch. Kay said got to do it, got to do it. It was only, you know, 20 more K of code at most, well, it actually turned out to be 12 more K of code on top of the 400K code. It wasn't that big a deal, and once Kay said it, it was very obvious, but we'd always wanted to do a low end operating system, and we had specs for low end operating systems, and we knew we were going to do one up on 16-bit.

Basically, what we wanted to do was one that was more like MS-DOS 2, with the hierarchical file system and everything, and the key thing was, my saying look, we can come out with a subset first, and just go upwards from that. Plus Paul saying that he thought he could go work something out with Seattle Computer Products, and have that as a foundation, and that's what made us say, hey, look, we can do this thing. And it really wasn't the hard part of the project. I mean, I was the guy, who when something was in trouble, would come in and review the code, and just sit there and bang out code on the thing, but I got involved with BASIC, Fortran, assembler, there were big challenges with all those things. And the reason I didn't get involved in DOS was those guys were on top of it. I literally never had to go thru the source code or anything, and I don't even remember the degree to which Paul was involved. He was somewhat involved at looking at the source code, but 12K, I mean, Jesus... I mean, Excel is 380K, that stuff is not linear, it's on the order of the square as far as complexity of testing and stuff like that...

Q. Did you remember what was the date of that...

MSC 00567042

September, 1980. Maybe early October, it's possible. It's not easy to remember stuff like that. There are parts of my life I

CONFIDENTIAL

RBC 001043

remember really well. like if you've ever been in a court case where you had to review everything that happens, its amazing to you how you can't do it, but if you're in that court case where you're reviewing it, then it all gets structured and every month you have some event, and I actually, I can tell you when it was... (goes to calendar)

Q. I'm amazed at how good the guys have been remembering..

MSC 00567043

CONFIDENTIAL

RBC 001044

It's such a classic here...oh, I'll get absorbed in this...I've got to close my eyes and pick a page...I was wrong, we didn't meet with IBM until...no, that was the 2nd meeting, ... Ok, it was, it was August, ... August 21, ...

Q. You're talking about the first meeting with IBM now?

Yeah, the first meeting was ... August 21 we had an IBM meeting, and we kept meeting with them. We met with them August 28, and I went down there, and ... September 11 we met with them, and I went to Europe...

Q. You can tell something happened because those-pages are all blank there...

It must have been September 29, which was a Sunday night. Cause Kay came in that Sunday. It could have been the 28th or 29th, but I'm pretty sure it was the 28th, I'm pretty sure it was a Sunday night. It's funny. This was about the time Intel was showing us this thing called the 432, and I was giving Intel such a hard time about that...and Fortran was so messed up...and see, as this thing gets into November we're just into that IBM stuff in such a big way.

Q. I have a sort of...back to the future...one of the things that occurs to me is there's some really interesting new technologies looming on the horizon. One that comes to mind is optical disks that is in a way several orders of magnitude different from what's around, both in terms of capabilities, storage capability, and also in terms of being multimedia, and I wondered if you had any thoughts of its relation to MS-DOS, if it's more or less just another kind of file-based system to add to it, or whether the facility's standing capabilities might really redirect how MS-DOS develops in the future...if there's been any thought on that, or if you have any idea...

Over time you want to get data independence in the sense that you don't know where your data is, and it's just up to the operating system to find it, not just in a physical location on the disk, but also on various disks, and across a network on various machines. We gotta get very involved in that, location independence for data... and we do have to get new file systems that can deal with very large devices, and we have to be able to support more than one file system at a time, right now we just have the original FAT file system, and we will over time have multiple file systems. The file system that you'd use for a large thing like that is not the FAT file system. We think files up to 2 to the 32nd are adequate for the foreseeable future, but this notion of data types embedded inside the files, DOS won't be getting involved in that in the near future.

If you want to talk about the CD ROM (?), I'll explain to you how Microsoft will, and even Windows will, but not DOS...

DOS is a thing that you've gotta put stuff in that everybody's

CONFIDENTIAL

RBC 001045

MSC 00567044

going to use very, very heavily...why put something in an operating system...I gave a great lecture about this several years ago when we were starting, when we were doing MS-DOS, it was my (Rosinform ??) thing where...if we look back, maybe we can find it ...OK, here's the appand here's the operating system and here's the hardware. What mediates what you should put in here versus in here? That asks you what's the job of the operating system. Well, the job of the operating system is first of all is to virtualize interfaces. Now that sounds like a bunch of gook -- why virtualize interfaces? OK, you've got an environment where the hardware is evolving and getting a lot better, and the software is evolving and getting a lot better, but the most important thing as far as the user is concerned is the soft revolution. He wants to continue using better applications. If the software is married to the specifics of the hardware, then you freeze hardware innovation, because nobody's taking advantage of it, and then you get this really abrupt change when finally some applications go to something, but it really messes things up. So the idea of the OS is to allow these people to do things like make bigger disks without changing things like 1-2-3, for the bigger disks. In the case of the disk, hey, everybody calls MS-DOS, nobody goes direct to the disk. We did a great job on that. In the case of most other things, it's a problem.

(Jim Beley leaves session to go to other obligation.)

So you want to have the ability to let new applications be introduced and new hardware be introduced independently of each other, so if you've got the right virtual interface in the OS, it's a layer that insulates that does that. That's really fantastic. The other thing is that the OS has to manage resources on ... if you have two applications here, they're both trying to get at memory, screen, CPU, and disk resources, and it's up to the OS to decide who's going to get those resources, and to do the appropriate interlocking, so two people aren't on the same printer at the same time, and the same com port where you've got both file applications.

And the final thing you want to put in the OS is... if there's some piece of code that almost every application has stuck into it, and you'd prefer not to make application writers have to redevelop it, and maybe do it wrong, and maybe not follow the standards, if it's so common then you almost put things in the OS like a shared subroutine library that are just there for everyone, you can always assume are there. And in this last category, you gotta be real careful, because the fatter this thing is, the worse off it is. It's true that the cost of fatness is, thank God, very, very low now, in terms of the cost of the actual RAM chips, it's not really a problem. The key thing that's pushing against fatness right now is that 640K amount of memory that's available in the address map of the 8086 IBM PC, that's what's really pushing this stuff down. So that makes us want to keep the OS really small.

One thing people might not know about MS-DOS is that every bit of

CONFIDENTIAL

RBC 001046

MSC 00567045

it is written in the 8086 native code, and we've never considered moving the thing out into a high-level language, because if there's anything that's worse (than hex?) for trouble for putting into native code, this is it.

And so, to get involved in things like audio and video formats is a little tenuous. I mean, Windows is an attempt to get involved in a video format and video interfacing in a big way. We have to study new CRT chips and make sure that our Windows virtual interface allows an application to do what it wants and yet take advantage of the evolution of video hardware.

So, the real answer to your question is, we will allow multiple file systems, they will be will be adapted to the large file structure, we will be more than just a file system. We will have data independence across machines. When you open something you don't have to give a machine name, slash, a directory, and something else, you just name data and it's sort of up to us to find where it is, independent of the name. And, and not stuff like audio and video...but if things like (write-once??) media really do become popular, then the type of file structure used for that is very unique. For read-only media, you want a file structure that's very unique, and Microsoft is actually pioneering what we think will be the standard for CD ROM logical file format---that's an on-going thing, in fact there's a meeting going on today on that.

Q. Does Windows run on version 3?

Yeah, Windows doesn't require version 3, it runs on v. 2 or v. 3, it runs on 2.0, 2.11, 3.1, well whatever... no problem. It's easier for us to keep Windows in sync with various versions of DOS than it is for other people to keep other things in sync with various versions of DOS. It's very helpful to us in terms of doing things like foreground tasks switching efficiently.

Q. I think you've covered everything. We don't have any questions. I guess the last thing to say is, you know, anything else that you have in mind that you think would be pertinent or useful, appropriate, for a history of MS-DOS?

I think I have so many old speeches in here. The problem with my speeches is that I never write down...there it is...this is this (Rosinform?) thing... I don't know, I don't know how valuable that stuff is...there's these conference notes from the (Rosinform?)..

MSC 00567046

Q. I haven't heard that term before? Yeah, what was the Rosinform?

It's a yearly conference, industry conference... these are all...half these speeches in here are MS-DOS speeches, I was promoting MS-DOS for so long ... In the Rosinform they always do transcripts, Natalie will have it in the library. And one of

CONFIDENTIAL

RBC 001047

those years, all I did was an MS-DOS speech, which talked about why we did MS-DOS. (Looks thru papers) That's pretty good if you can read it.. it's all about OSs and how some people believe that OSs will never do stuff right...for some reason I throw in all this wierd stuff about graphics at the end, which really isn't MS-DOS, but we stuck it in Windows...I don't know...Just for the heck of it, that's a speech that I wrote out, the only one I ever wrote out in my life, and this is the slides I gave with the speech, but the easiest to read in terms of the philosophy of MS-DOS, has gotta be to go back to those old ... I kept everything. I'm out of control.

Q. This is quite legible...

You can keep that...

Q. Do you want us to send it back to you, Xeroxes and all?

You might as well send it back to me just for fun...don't think that there's any rush to turn any of it back... There's nothing else, that's it.

See, I was not a developer of DOS. Very few people were, that's what amazes people. It's not that big a deal. Yeah, after DOS 3, it's a big deal. It's a big deal. Whew! With terminate and stay resident tasks doing the stuff that they're doing trying to make that stuff run and the subsystems that we have and trying to take advantage of 286 protected mode, it's a big deal. It's complex, it's hard. You know, some people say to me, will somebody write a clone of MS-DOS some day? Maybe they will, but good luck to them! I mean, this thing is complex as far as trying to make stuff run on top of it. Very, very, very, very complicated. It's a \$65 dolla- product...I mean what if it was \$40, what if it was \$36, how many people are willing to save money to get an imitation of DOS? Someday we may see, but so far... drrr has attempted to do things that are DOS compatible, but they haven't even come close yet. They're always a few generations behind, but they never get the thing exactly right. That's one of the funny things about DOS; we always try to do things cleanly, but to the degree that things aren't done cleanly, it almost makes it harder to ever mimic the thing. OH, you look at the thing some days, and you think, oh, maybe we intentionally created the thing this way. We did a good job of making it hard for ourselves to upgrade it. The people who work on DOS, compatibility is a religion. To go sit down and write a new operating system that's not at all compatible from scratch, you know, we could take a fifth of the MS-DOS team and give them two years, and they could write a from-scratch operating system that would be as good technically, and probably better technically, you know, because they'd have this clean slate of paper, than doing the new stuff... and then they wouldn't have to do all that new stuff. But the world wouldn't care, they really wouldn't care... There used to be so many more operating systems in this world than there are now, I mean, look at this Unix

MSC 00567047

CONFIDENTIAL

RBC 001048

thing. I mean, how successful is UNIX? Well, it's about 150,000 work stations, of which we have over 100,000...and you compare that to DOS, and by the time your book gets out will be on the order of 4,000,000. It's probably on the order of 3,000,000 now. Add it up and figure it out. MS-DOS, there's not the slightest doubt in our minds, that all the popular home computers will be MS-DOS-based machines two years from now. It hasn't been true, it's been Commodore 64, and Apple II, and stuff like that, but the premium for making a 16-bit machine is like zero nowadays, so why not make it an 8086 MS-DOS machine? Well, it will be.

Q. How do you see, like the Macintosh, as the exception of that?

I will be the only exception, and it will be an exception because there's room for two standards and Apple's a critical mass. But that will be it as far as something that's truly incompatible. We think the Macintosh is very nice.

Q. Obviously. I guess we'll go back and transcribe these tapes...

Gates - So how does this work? You're doing certain chapters, and other people are doing other chapters?

Q. Well, the writing is divided ... right now there are five major parts of the book. The first part is the history section, and the second part is the user commands, and the third part is kind of a description of the system point of view, sort of just the structure of MS-DOS, and the fourth part is the detail, the system calls, and the fifth part, which is in question at the moment, is just a systematic treatment of version by version changes. Different people are doing those different sections. Howard (?) and I are doing the first part, and we're doing the edit of the whole book to make sure that everything is in the same voice and all of that...

Gates - Really? Great.

A lot of the design ... there's some real problems with user manuals, so we've done some signifi

Well, the hope here is that everyone who develops MS-DOS will find this indispensable...at least that audience...maybe some of the hangers-on who'd like to have definitive works about things that they may never read might want to have the thing sitting on their bookshelf as well. Does stuff about ... I think the version to version stuff is pretty interesting...I hope we can do something there, even if it's not like those other 4 things...because people, you know, nowadays you want to be compatible with version 2 and version 3 both, you'd sort of like to take advantage of the version 3 stuff, and you'd like to

MSC 00567048

CONFIDENTIAL

RBC 001049

understand what the different is, and version 2.01, 2.11, what the hell is that, and I think...

Q. I think the question at the moment is not desirability, but resources...

Page count too. We're talking about 1200 pages now, and we're pushing that all the time...

Gates - I think it's important not for it to fall below 1000, or else it won't have the mass that it's supposed to have.

Q. I don't think that we'll have a problem with that.

Yeah, there's a lot of neat things that are being included that'll expand it, like the flowcharts for each call, tables, use of graphics..

Gates - It's a great project, I'm real enthusiastic about it.

Q. We really appreciate your support.

MSC 00567049

CONFIDENTIAL

RBC 001050