

OS/2 to open new capabilities for PC developers, users

With an installed base of about 6 million machines, MS-DOS has become business's standard PC operating system. Its creator, Microsoft Corp. of Bellevue, Wash., has now developed its replacement, OS/2, as part of a joint development with IBM. The new OS/2 is aimed at the Intel 80286 processor, which allows multitasking through a protected mode and faster processing. It also removes the 640K-byte limit on addressable memory.

In an interview with Assistant Editor Galen Gruman, Mark Mackaman, Microsoft's OS/2 product manager, discussed the new operating system.

Q: With OS/2 soon to be available, what about the millions of MS-DOS users?

A: The two systems will coexist simultaneously. We don't see OS/2 becoming dominant for two to four years. But it will eventually be dominant in the business operating environment.

Q: Because of IBM's proprietary MCA bus architecture, will you have to abandon support of current hardware?

A: No. At this point, we're using the IBM PC AT as our standard. It's on every desk in business. PS/2 is, for now, nonstandard. It would be crazy to use the PS/2 hardware as the standard and require developers to adapt the operating system for the AT. If PS/2 becomes a standard, we will change the OS/2 developers' kit to use the PS/2 hardware as the base configuration.

Q: There has been criticism that OS/2's compatibility box will run only one DOS application at a time and that it works slowly. Why the limit?

A: I don't understand why that's a criticism. Under DOS, you can only run one program now, so why require that OS/2 handle more than one?

Q: Perhaps people want to take advantage of OS/2's advanced multitasking features without giving up their current software investment.

A: Of course. Maybe they also want more memory than 640K in the compatibility box. But then they really want a DOS Plus. Our intent is to provide a clone of the DOS environment that you can use from OS/2. If someone is going to use only DOS applications, then don't buy OS/2.

Q: If you're limited to one DOS program, what about the terminate-and-stay-resident programs like Sidekick?

A: Whatever you can do in DOS now, you'll be able to do in the compatibility box. But you really don't need TSRs because OS/2 has multitasking. You can just call up a program in another window. We put in the hooks for TSRs [in MS-DOS] because people wanted some sort of multitasking ability in DOS — but DOS wasn't really designed for it. I've been surprised how creative people have been with them.

Q: What should people be warned about when running DOS programs in the compatibility box?

A: When you leave the box, the DOS program stops. It's in suspended animation. It doesn't run in the background, even though when you go to the box, your OS/2 programs continue to run in [the 80286's] protected mode. This could cause problems with time-sensitive applications. But we had to do that because DOS writes directly to the screen and we couldn't tell it to stop writing while it was in background.

Q: Why will OS/2 benefit developers?

A: DOS has reached its limits, and OS/2 removes the 640K RAM and multitasking barriers.

Q: Do you think developers need much more than 640K, or are they just using as much as is available?

A: A lot of programs are already up against the 640K ceiling. Look at Symphony; it has a lot of capabilities. They're not just wasting memory. We can't extend DOS because that would knock a lot of programs out; there wouldn't be enough RAM left. OS/2 will let them address 16M bytes of RAM, and with its virtual-memory management, the amount of memory is unlimited.

Q: It sounds like the opposite of a RAM disk. Instead of using memory as a disk, you use a disk as memory.

A: Yes. And it can handle up to 512M bytes per application, theoretically.

Q: Theoretically?

A: Well, we don't have any programs that big to test! But I'm sure someone will come up with one. Look at AI. Some programs take a megabyte just for the code. Some companies, like Gold Hill, have come up with some tricks to make it work on an AT, but 640K just isn't enough. I think that's why AI hasn't done any better.

Q: Lately, with the new Mac II and the PS/2 systems, it seems that Apple is getting more like IBM and IBM is getting more like Apple. Do you see the two approaches merging?

A: The idea of a common user interface makes sense. Common interfaces make training a lot easier. People now have to learn different interfaces for different products. And it makes sense to have a standard programming interface.

I like to use the car analogy. All the car makers put the gas pedal on the right and the brake in the middle. If there's a pedal to the left of the brake, you know it's a clutch. "P" means park and "D" means drive. But you can tell a Chevrolet from a Ford. Each car has a different style even though they follow the same conventions.

Q: I suppose a developer could ignore the tools provided, and it would work in the environment but would look any way he wanted.

A: They can ignore all our tools and design the interface from scratch. But we hope they don't.

Q: You're not restricting the developer from doing what he wants as much as you're providing standards. Is that the difference?

A: We provide the tools the developers use. We hope they will put a command bar at the top and have pull-down menus because that seems logical, but they can put them at the bottom and make them go up. They can do it any way they want.

Q: As someone who came from mainframes, I find graphics interfaces cumbersome for some applications, like word processing and spreadsheets — although I can see how they help new users.

A: I had the same reaction when I started using graphics environments. But they do more than help new users. I really think they open a whole new level to the user if they're presented right. Look at word processing. You can see the type on the screen. You can get your headlines to appear like you want, not wait until it's printed. That opens up what you can do.

Well, the same will happen for other applications. Someone will find ways to use the new capabilities to bring the work to a higher level. We're in a Model T era now. Remember, the industry just started in the late 1970s.