

September 28, 1988
Vol. 1, No. 16

Program Management: Making OS/2 Useable

by Mark Cliggett

Over in the Apps division, Program Managers write the specs for the applications, ensure that the program is being tested, and in general, ensure that the project goes as smoothly as possible. Here in the monolithic Systems group, the same duties have been divided over several different groups. Program Management in our case is equivalent to writing the spec - ensuring that OS/2 is and will remain very competitive. This falls in three main areas: making sure that ISVs are getting the functionality they need in a form that is useful, understanding what we should be doing in the future to keep OS/2 competitive, and finally understanding what other key parts of MS are doing.

Getting a lot of great applications up and running on OS/2 is far and away the most important factor in the success of OS/2. Without applications that take advantage of its functionality and give users a reason to switch to OS/2, OS/2 will fail. There are three things an ISV considers when looking at OS/2: Is the functionality there? Does it perform well? How usable is the functionality?

Functionality-wise, OS/2 has a lot to offer. The developer coming from the MS-DOS world suddenly has plenty of memory, multi-tasking, and six zillion flavors of IPC - in other words, a real operations system. The developer coming from the mini or mainframe world has most of the familiar OS/2 functionality, a friendlier user interface, and a big graphics library - all on a machine

devoted entirely to one individual. It is not surprising then that feedback on how much functionality OS/2 provides has been great. People attending the OS/2 seminars were very excited by it all, and one of the weeklies call our SDK "a programmer's dream".

Dan Hinsley is responsible for the performance of our product. However, since it affects ISV's ability to use the system, we also have an interest in this. The feedback so far on performance has also been mostly positive. Work done here shortly before v1.0 shipped squelched the idea that the base is slower than MS-DOS. The recent feedback from PM ISVs is that PM performs at least as well as Windows - no mean feat given the additional cost of serialization and running in protect mode. There are however a few areas which we should be trying to improve - scheduling, vector fonts, and overall memory usage, among others.

By far the most important role we have in watching out for the ISVs is trying to improve the usability of our API. We have tons and tons of great functionality, but we do a poor job of making it easy to get at. Three examples that immediately come to mind are PM apps, IPC and memory management.

Most people here know how hard it is to write a PM app - the classic example is the 200 line "hello world" program. Sure, once you have those 200

lines, you can write a lot more sophisticated app with little extra work, but if you're looking at PM (or Windows) for the first time, those first couple hundred lines are extremely difficult to get right. We are looking at everything we can do to ease the process - lots of sample code, documenting the 100 functions that you really need to know to get started, a resource editor that generates skeleton apps automatically, higher level api... literally anything that might make it easier.

Most people here know how hard it is to write a PM app - the classic example is the 200 line "hello world" program.

The first thing that ISVs picked up on at our seminars is that we provide 79 different ways of doing inter-process communication. Many of the methods sound related (base queues and PM message queues), some conflict (PM and signals), and in general there is a lot of confusion about when to use what. Again, things we're considering sample code, additional all-encompassing api (for example, a PM messaging api which transparently works across a network and trimming down the published api (in the 386 version).

A third example of hard to use apis is in the area of memory management. We provide several flavors of global memory management (DosAllocSeg, DosAllocHuge, named shared memory, etc.) and two sub-allocaters (one in the base, and one in pm). The limits on selectors and the performance problems associated with lots of small objects may make these necessary. However, we've essentially just passed these problems on to the ISV. They have to decide whether an object is big or small to know which api to call, and then have to remember how each object is allocated. Most ISVs would be happy calling DosAllocSeg for every object. Indeed, many less sophisticated ISVs probably do exactly that, but would be a lot happier if we managed the small objects for them. There really should be only one way of allocating memory,

particularly on the 386.

Mike Hyman has already raised a number of 1.2 dcrs to make some of our graphics api easier to use. In most cases there is no new functionality involved. Just a few very simple, slightly less flexible calls which make existing functionality easier to use. If anyone has ideas about other things we can do to make things simpler, please talk to one of us.

Another important function of the program management group is acting as the interface between operating systems and other parts of Microsoft - Languages, Apps, the net group, and Nathan's Advanced Development group.

Languages will provide the bulk of the tools that ISVs use for developing OS/2 applications. Part of making it easy to write OS/2 applications involves understanding what our Languages group is doing with respect to OS/2. They have a lot of environments to support (MS-DOS, Windows, OS/2), and a lot of pressure from competitors. We need to make sure that OS/2 is getting the attention that it needs, and that ideas to improve things like the Dialog Editor get acted on.

We have tons and tons of great functionality, but we do a poor job of making it easy to get at.

Our Applications division is a great place to get feedback from a real ISV early enough that we can fix any problems that arise. My favorite example here is when the Excel group found that there was a limit of 255 code segments and resource. Given that they have about 300 (mostly resources - dialog boxes, strings, etc.) under Windows, this was a real problem. It turned out that the limit only needs to apply to code segments, and with a small amount of work the problem was fixed. If they hadn't communicated to us, or we hadn't followed up, we would have shipped with a pretty embarrassing problem.

Distributed processing and groupware are two areas that people believe will be very important in the next few years. Our network group is trying to make these applications possible. To do so, they'll need support from the operating system. We have not done a very good job so far of giving them this support. Features they depend on do not make it into the plan and APIs are designed that do not network well. I'm sure they can come up with additional examples. We will be working to improve the communication and end this sort of trouble.

Within the OS/2 group itself, everyone probably has a list of 50 things they think could be better, not just within their own code, but throughout the system. People within the group know the details, know what's weak, and what could be half the size and 10 times faster. Often though the schedule or inertia make it difficult to get these things fixed. We are not very good though on following up and getting them fixed in the future. I would like the program management group to be a focal point for these kinds of problems. If you find something that you think could be better, but cannot get it fixed yourself, at least send us email explaining what the problem is and what you think needs to be done. We will keep track of them and try to organize the work to be done at appropriate times in the future. For example, if the scheduler needs to be reworked to handle more than 255 threads, then that is the appropriate time to think about fixing the scheduling algorithm too.

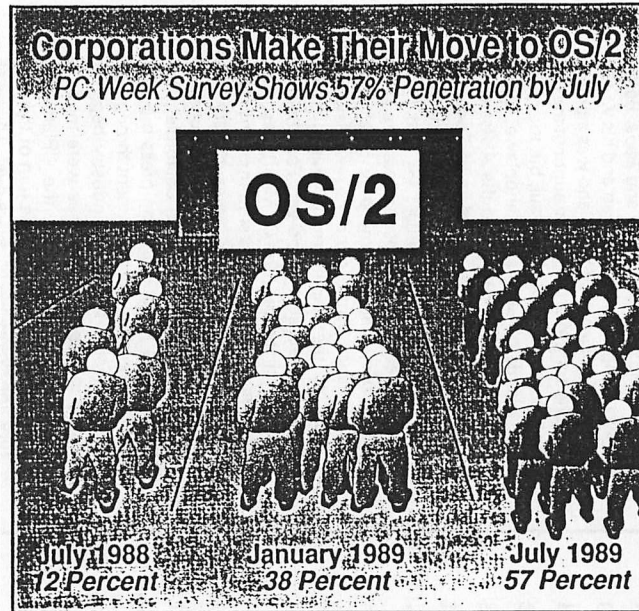
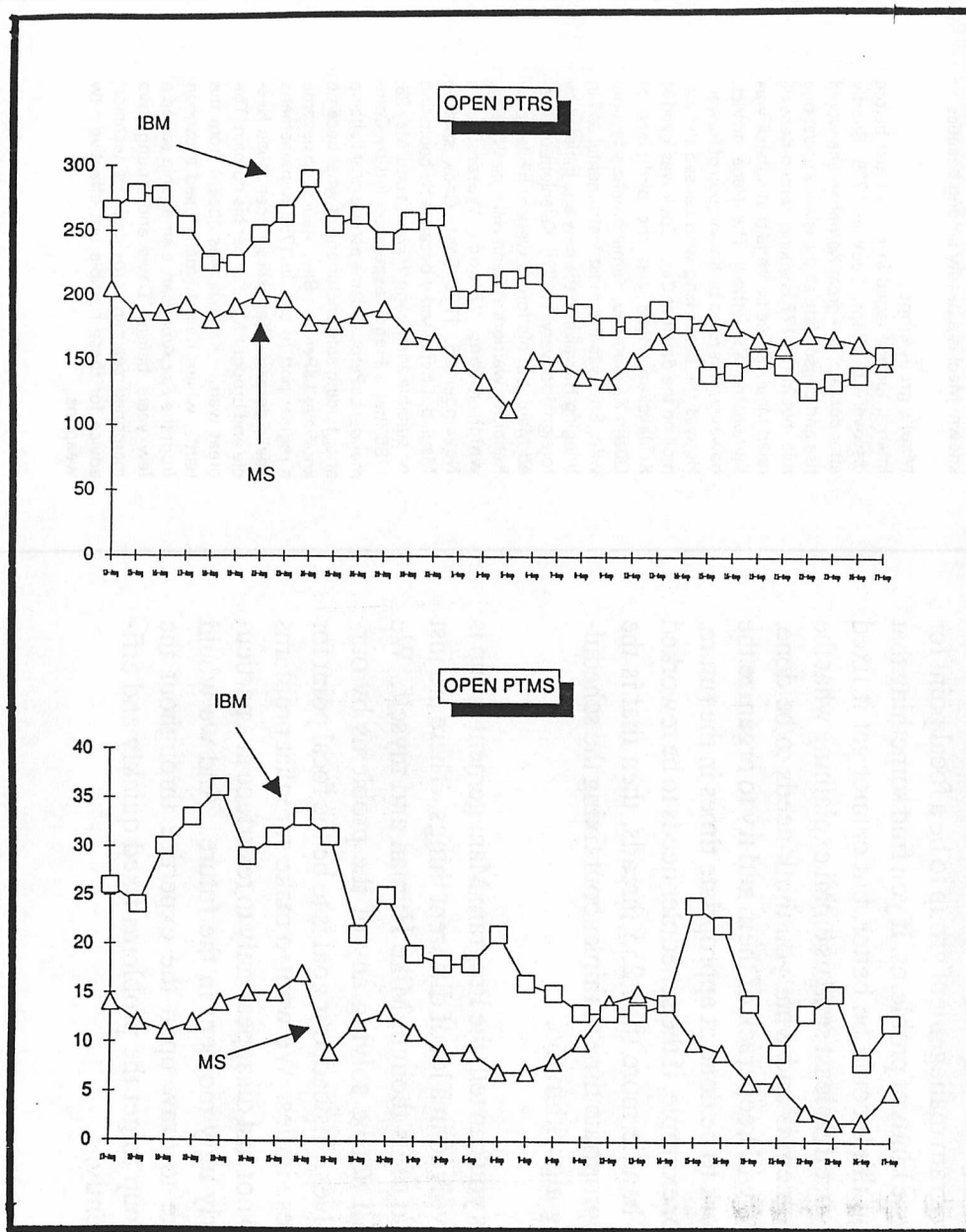
As you can tell, the Program Management group is involved in a lot of different things. There are just two of us though, Mike Hyman and myself. We will not be solving any of the problems by ourselves. Instead, our goal is to be the focal point for these issues. We want to make sure that problems do not just disappear only to resurface at a particularly ugly moment in the future. And we would like to draw upon the expertise throughout the group to get the problems fixed quickly and efficiently.

The *Kumulipo* sings that Maui was the second island child of *Wakea* and *Papa*. Before the coming of the white man and his written record, it's clear that the island was a powerful kingdom. Wars raged throughout the land and kings ruled not only Maui, but the neighbor islands of Lanai and Kahoolawe. By the 16th C., a royal road called the *Ala'oa* encircled the island and signified unity. Today, on West Maui, the road is entirely obliterated: only a few portions remain on East Maui. When the white men began to arrive in the late 1700s, Maui became their focal point. Missionaries, whalers and the new Hawaiian kings of the Kamehameha line all made Lahaina their seat of power. For about 50 years, until the mid-19th C., Maui blossomed. Missionaries built the first permanent stone structures in the islands. An exemplary New England-style school at Lahainaluna attracted students even from California cities. Here, too, a famous printing press brought not only revenue but refinement through the written word. The sugar industry began in secluded Hana and fortunes were made; a new social order under the "Plantation System" began. But by the turn of this century, the "glory years" were over. The whaling industry faded away and Oahu took over as the central power spot. Maui slipped into obscurity. It revived in the 1960s when tourists rediscovered what others had known: Maui is a beauty among beauties.

Maui's great kings

Internal turmoil raged in Hawaii just before discovery by Capt. Cook in 1778. Shortly after contact, the great Kamehameha would rise and consolidate all the islands under one rule, but in the 1770s a king named Kahekili ruled Maui. (Some contend that Kahekili was Kamehameha's father!) The Hana district, however, was ruled by Kalaniopuu of Hawaii. He was the same king who caused the turmoil on the day that Capt. Cook was killed at Kealahou. Hana was the birthplace of Queen Kaahumanu, Kamehameha's favorite wife. She was the most instrumental *ali'i* in bringing Hawaii into the new age initiated by foreign discovery. In 1776, Kalaniopuu invaded Maui, but his forces were annihilated by Kahekili's warriors at Sand Hill near Wailuku, which means "Bloody Waters." On November 26, 1778 Capt. Cook spotted Maui, but bypassed it because he could find no suitable anchorage. It wasn't until May 28, 1786 that a French expedition led by Commander LaPerouse came ashore near Lahaina after finding safe anchorage at what became known as LaPerouse Bay. Maui soon became a regular port of call. In 1790 Kamehameha finally defeated Kahekili's forces at Iao Needle and brought Maui under his domain. The great warrior Kahekili was absent from the battle, where Kamehameha used a cannon from the *Fair American*, a small ship seized a few years before. Davis and Young, two marooned seamen, provided the technical advice for these horrible but effective new weapons.

GOOD NEWS DEPARTMENT



Survey Shows OS/2 Is Gaining Momentum in U.S. Companies

By Richard Stromer

The persistent wait-and-see attitude of most corporate micro managers toward OS/2 appears poised to give way to wide-scale, if cautious, implementation.

According to data from a recent PC Week survey, more than one-third of U.S. micro installations expect to begin migrating PCs to OS/2 by January 1989, while nearly six out of 10 plan to do so by July 1989.

That survey, conducted in July among members of PC Week's Corporate Advisory Panel, also found that one-tenth of all 80286- and 80386-based PCs are projected to be running OS/2 by early 1989. By mid-1989, the share of such high-end micros running OS/2 is expected to exceed one out of every six 80286/80386 units.

Comprising more than 500 individuals who play key roles in their organizations' acquisition and implementation of technology, the PC Week Corporate Advisory Panel is designed to represent a reliable cross-section of U.S. companies and institutions employing PC technology.

Survey Sites Say 17% Of High-End Micros Will Run OS/2 by July

Continued from Page 1

In general, the survey's results indicate that OS/2 will most likely be integrated slowly with DOS, Unix and other operating systems. As of late July, only 12 percent of panel sites had installed OS/2, and only on an average of 3 percent of their installed 80286/80386 PCs.

When panel members were asked to predict their OS/2 usage six months in the future, the penetration rate for OS/2 showed an increase of more than three-fold, to 38 percent. Overall, a projected 10 percent of all 80286/80386 micros are expected to be running OS/2 by the end of January 1989.

Finally, when panel members were polled on the likely status of OS/2 at their sites as of July 1989, 57 percent told PC Week that they expected to have installed OS/2 by that time. By then, an average of 17 percent of such sites' high-end micros are projected to be running OS/2.

The industry group with the highest current penetration by OS/2 is the sector comprising commercial data-processing services. Thirty percent of these businesses have already migrated micros to OS/2. The next-highest current OS/2 penetration occurs among educational and medical institutions, of which a combined 23 percent have already begun using OS/2.

Kollsman Co., a division of Sequa Corp., located in Merrimack, N.H., is an example of a data-processing company that currently has OS/2 installed. Al Carrier, project leader supervisor, runs packages to download and upload from mainframe to PC. He said he will probably increase OS/2 use on

other machines and believes compatibility "will get better."

By early 1989, the single-highest use of OS/2 is projected among manufacturing companies, 61 percent of which expect to be using OS/2 within six months. This data is especially interesting, as only 9 percent of manufacturers had migrated PCs to OS/2 at the time of the survey.

A similarly sharp upswing in short-term OS/2 adoption rates is anticipated by government agencies, increasing from 12 percent of such agencies currently to 59 percent by January 1989.

Stacy Newman, microcomputer specialist for the Congressional Budget Office in Washington, will not make a decision about OS/2 until after her department evaluates it.

Top OS/2 Users

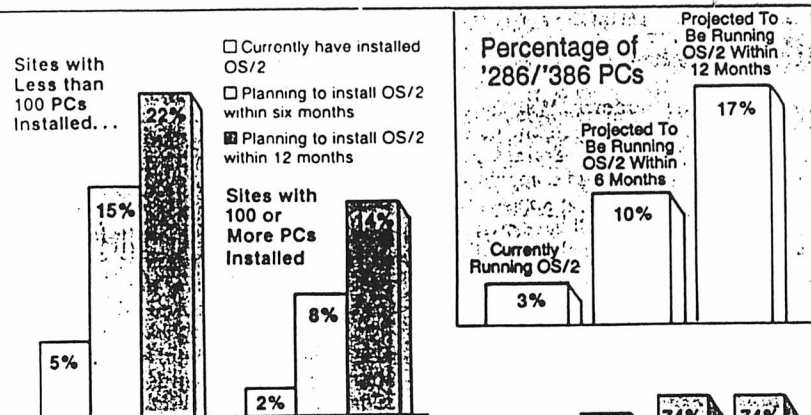
Both government agencies and manufacturing firms are projected to continue among the top sectors using OS/2 by mid-1989, with 74 percent of the former and 71 percent of the latter expected to do so. Other major adopters a year from now are expected to be both educational and medical institutions (72 percent) and businesses in the combined agriculture, mining, oil and construction sector (74 percent).

Regardless of either industry grouping or time frame, larger sites—those with at least 100 PCs installed—were considerably more likely to have OS/2-related plans than smaller sites.

Currently, only 6 percent of small sites have migrated to OS/2, compared with an average of 18 percent for large sites. Although 52 percent of large sites expect to be using OS/2 by early 1989, only 30 percent of small sites plan to do so. Finally, 72 percent of large sites expect to have begun migrating their '286 or '386 PCs to OS/2 by mid-1989, compared with 51 percent of small sites. ■

Research assistance provided by Laura Cooper McGovern

Migration to OS/2 on the Rise ...



Percentage of Sites Planning to Use OS/2 Within the Next 12 Months (By Industry Groups)

