I HAVE TO LEARN WHAT?!

Many of the people in chemistry today who have used computers for some time grew up in a world in which we dealt with a particular piece of software, prepared input for it, submitted it to a computer somewhere and awaited the results. This was the 'batch mode' world.

Our contact with large computers was really very limited. We did not possess detailed knowledge of the architecture of the computer or how it worked. The programming language we used was FORTRAN, which was completely oriented toward the types of problems we worked with.

Certainly, all of us realized that such things as assembly language programming existed, but we had no need to learn how to do it. We were also aware (in varying degrees) that something called an operating system was actually controlling the overall activities of the computer; but, again, our programming language gave us virtual independence from the thing called the operating system, so we never bothered to learn all that much about it.

On those rare occasions when we had brief problems which could readily be identified as coming from the operating system level of the computer, we were invariably able to take our problems to someone called a systems programmer. We never developed any systematic appreciation of what this person knew or actually did but, often as not, our problems went away.

As time went on, our next memorable computer experience was with the IBM-PC and DOS. This machine made us intimately aware of the existence of the computer's operating system – DOS. We were forced to delve more carefully into those aspects of computing which heretofore had been hidden from us in a large central-computer environment. We learned new things, we successfully dealt with the system-level problems, and many of us became more self-assured in our encounters with computing. We may also have fallen victim to a touch of overconfidence...

The DOS operating system can be described as a minimal operating system, i.e., it provides a minimal set of operating system services but creates a useful computing environment. It is a well-conceived system which can be utilized by most application-oriented users with very little difficulty. The designers of DOS could possibly be accused of making it too easy for us.

At this point we need to pause and look carefully at the computing environment we are about to enter. We are moving up from our simple IBM-PC to something called a workstation. The workstation rarely exists alone. It is usually connected in some sort of network to other workstations and PCs and usually has access to a large-scale computer somewhere in the world.

The workstation itself is far more powerful than a PC. Many of the workstations that can now sit on top of one's desk have more raw computing power than the central computers of the 1960s which served whole institutions. For example, the IBM 7090, the workhorse of the early 1960s and one of the first solid-state computers, had a clock time of one microsecond. This sort of performance would be an abject embarrassment in today's world. We have come a long way in a few short years, but not without a certain cost.

Today's workstation is more like a large-scale computer than an IBM-PC. To adequately take advantage of all this computing power, each workstation contains a full-blown operating system. This system is not at all like simple DOS. It is, rather, like that of nonsimple central-computer

0920-654X/$ 3.50

operating systems. For the individual to make full use of his workstation, he must as often as not come to grips with its operating system.

Upon taking delivery of a new workstation one has a machine which is analogous to a new baby. It is truly a 'tabula rasa'. One usually receives a tape cartridge and detailed – extremely detailed – documentation about how to carry out what is known in the programming trade as a SYSGEN (generating of the system). Previously, SYSGENs were carried out on the central computer by systems programmers. This is one of the things they did.

It is necessary to immediately identify one person by the account name SYSTEM. The person with this designation will be the only one with the authority to make critical additions and modifications to the operating system at its lowest, most primitive levels. This person will also be the one to whom any other users will come with system-related problems. In effect, the person with this designation *is* the systems programmer – and there always has to be one.

The message here is rather simple: When you buy a workstation, you should give some thought about how much you will enjoy becoming a systems programmer. For people at large institutions, it may be possible to have a single systems programmer who can service perhaps 50 workstations. Someone has to do it.

Very few workstations exist in isolation. Usually they are a part of a network and, ultimately, part of many networks by means of gateways which link these networks.

Here, too, the scientist/user is brought into contact with a class of problems unfamiliar to him. In fact, the sorts of problems exhibited by networks have largely been unknown to all but a small community of experts in large-scale telephone nets and such. The problems are of sufficient importance that major computer vendors have formed task forces to address them with as much haste as possible.

Yes, the workstation is an important addition to our ability to practice chemistry computationally. However, as you go into the workstation era, try to have someone on hand to help with the problems outlined here. A solid systems programmer is worth his weight in microchips.

Richard W. Counts
QCPE
Indiana University
Bloomington, IN 47405
U.S.A.