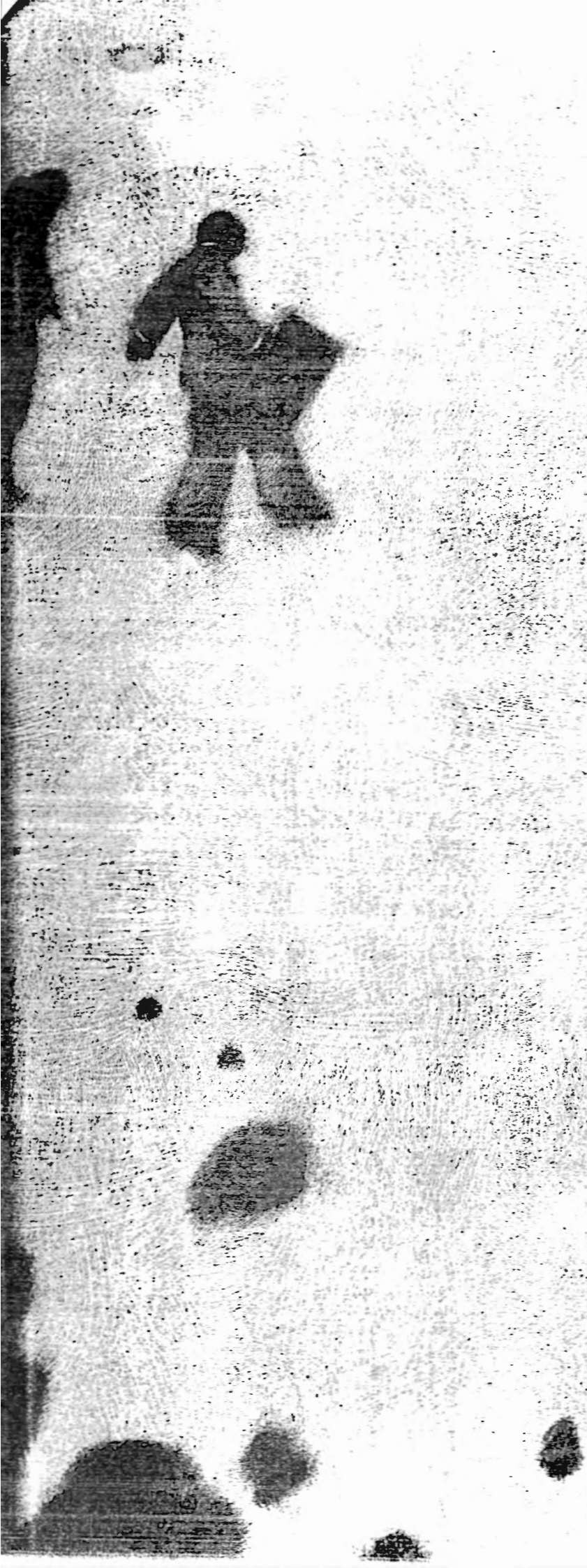

MULTICS USERS FACE THEIR MAKER

Impassioned proponents are fighting to keep their elegant operating system from being abandoned.

by John W. Verity

"The company's long-range product policy . . . is strongly oriented to the protection of existing customers' investments," states one of Honeywell Inc.'s annual fact books. Products "are designed to have a long, useful life and to provide natural growth paths for users and their ever-increasing data processing requirements."

"Bull," says a group of about 50 large Honeywell mainframe customers, and they don't mean that vendor's French affiliate. They've just been told what every user dreads hearing most—their operating system is headed to that big core dump in the sky, that it is about to reach the end of its "natural migration path" and perhaps never be seen again.

Indeed, these users say, unless a white knight arrives, their operating system faces a quite *un*natural, unnecessary death. Something so "elegant" and "beautiful" and . . . well, just plain "productive," they insist, should be kept going for its own sake. As those adjectives imply, they aren't grieving for just any operating system. They're mourning Multics, a system whose users are few but probably more devoted than any you can find. Unabashedly, they express the belief that theirs is simply the best operating system the world has ever seen.

In January this year, Honeywell told them that it had dropped

ILLUSTRATION BY ROB COLVIN

Even Honeywell officials concede that to know Multics is to love it.

previously advertised plans to deliver larger Multics hardware to them. A few more releases of the operating system are to be shipped for existing machines, the company stated, but there will be no further development of the special 36-bit hardware that Multics needs to soar. Unspecified "core features" of the 20-year-old system might show up in a future line of 32-bit minicomputers, but in effect, the Multics cherished by users is about to reach the end of the line. By this point, users are running out of floor space with current Multics hardware. Since Honeywell decided to "abandon" them, as they put it, they have nothing but trouble to look forward to.

The Multics clan is, to say the least, disappointed and angry. Users say they were not surprised at Honeywell's announcement because, as one puts it, they have "heard it all before." Even though Honeywell has long favored GCOS, its "mainstream" operating system, many say they still cannot believe a system so clearly superior as Multics is being left to die.

Indeed, to hear Multics users describe their neglected "orphan" system is to question the wisdom of those responsible for its marketing. Multics has had great influence on the very notion of interactive, personal computing. It has inspired operating system design in innumerable ways. The system's survival in a marketplace dominated by comparatively crude competition is a strong tribute to Multics' inherent quality, as despondent users now claim.

Even Honeywell officials concede that to know Multics is to love it. In a mainframe market dominated by IBM's batch-oriented MVS and its awkward Job Control Language (JCL), Multics' seamless collection of interactive computing, memory management, applications development, and, most notably, unparalleled security features is a paragon of efficiency. It creates what many consider to be the best timesharing environment around, better even than Unix, which it indirectly spawned.

"It runs rings around TSO and VM/CMS," says Roger Roach, director of operations and systems at Massachusetts Institute of Technology, referring to IBM's two most popular timesharing systems. "It's still state of the art," 20 years after being conceived at MIT with help from Bell Telephone Laboratories and General Electric Co., he says.

"Anyone who uses Multics becomes an avid follower," states John Hergert, a Multics specialist who consults with Cutler-Williams Inc., in Dearborn, Mich. "There is nothing in comparison."

From almost every aspect, Multics displays a technical elegance that no other operating system can match, proponents say. To begin with, 93% of its 2.5 million lines of source code are written in the high-level language PL/1, not in assembler or some other cryptic code. Programmers, moreover, have on-line access to all of that source code and can inspect and alter it to tailor individual functions to their liking. Each user's set of personalized commands remains private and does not affect other users. In addition, the system's many pages of documentation are available for on-line perusal.

"I've gotten really spoiled by having the source code on-line," says Rick Sendelbach, a senior systems analyst at Southern Co. Services Inc., Atlanta, which uses Multics primarily for engineering work. "You don't have to consult manuals."

POWERFUL SET OF COMMANDS

Another draw is the system's powerful set of commands. "They all adhere to global arguments," explains Robert Walker, an esteemed Multician at the Rome Air Development Center, a major Multics site in upstate New York. "They have the same form no matter where they're used and they won't blow up if you use an extra argument by mistake. In Unix, different commands work differently, but in Multics you can just try a command and see what happens."

Susan Rosenbaum, president of Practical Solutions International, Plainfield, N.J., and an early Multics developer, compares working in Multics to riding a bicycle: "Even if you've been away from Multics for years, it takes only a few minutes to get a feel for it again. Soon you're rediscovering all the things you thought you might have forgotten."

But it is what Multics enables in the way of rapid program development and operational flexibility that tends to make true believers. The system's sophisticated virtual memory scheme, dynamic linking facilities, and powerful stack architecture make for highly productive, top-down programming and prototyping of applications, users claim.

Multics' virtual memory is widely acknowledged to be the most powerful implementation of that technology in the industry. Each user is provided with a 4-gigabyte address space that can be treated as if it were all in main memory—in effect, disk files are viewed as extensions to main store. The virtual memory algorithm requires no user involvement. "It is never necessary to worry about memory" even when large data arrays or programs are being used, notes Karl Laubscher, director of Multics marketing at Honeywell.

Dynamic linking "eliminates another big worry," Laubscher says. Under Multics, modules can be developed independently and then run together on an ad hoc basis. Linking, or resolving procedure references, takes place on the fly, so to speak, at the moment each module is first called into action. Programmers need not (indeed cannot, without difficulty) initiate a link edit process such as most other programming environments require.

LIMITLESS STACK IMPORTANT

Just as important is the system's effectively limitless stack, which facilitates symbolic debugging of programs. Explains Walker, "If you're testing software and it blows up, you can freeze the environment and build a new environment on top of it. You can establish a reference point in the stack and probe the old environment. Once you've found your problem and repaired it, you can return control to the old environment and continue execution from where you left off." Unix machines are generally too small to permit that kind of debugging, he contends.

At a 1972 computing conference, Multics designers described the advantages of building their system in PL/1 and having source code available on-line. For one thing, these features helped each member of the Multics team become familiar with the entire system, not just isolated portions of it. Also, reported F.J. Corbato and J.H. Saltzer of MIT and C.T. Clingen of Honeywell, it was "easy to manipulate the many components of different versions of the system. Thus it has been possible to maintain steadily for the last year or so a pace of installing five or 10 new or modified system modules a day. Some three-quarters of these changes can be installed while the system is in operation. . . . Application users need not be interrupted while the software they are using is being modified." Many users claim Multics is still the most productive and robust development environment they've come across. Some, such as Accu-Ray Corp. in Columbus, Ohio, and Honeywell itself, use it for developing software for other mainframes, minis, and microprocessors. They take advantage of its robust set of programming tools that enables code to be executed in "rough draft" form: only the highest level of a program need be coded explicitly—subroutine calls can be left unresolved and written later. Moreover, Multics is said to excel in its ability to help routines coded in one language make calls to routines in other languages.

Ford Motor Co.'s Engineering

Many users claim Multics is still the most productive development environment they've ever seen.

Computer Center in Dearborn is particularly keen on Multics. Usage of its three systems—each with five processors—has been growing at a compounded annual rate of 35%. In an internal Ford white paper currently making the rounds of Multics sites worldwide, the engineering group describes the exceptionally good productivity it has achieved with Multics. For instance, only five system programmers are needed to support the three systems, which together handle some 700 applications and 1,800 relational databases, the paper states. Between 700 and 1,000 users log on to those systems during an average day, often with over 300 being on-line at the same time. To maintain one particular interactive application, averaging 100,000 transactions a day, requires only half the time of a single person each year. Ford has found systems programmers to be as much as four times more productive using Multics than they are using IBM's MVS.

Ford also finds Multics' open architecture quite valuable. Its three systems support over 150 different types of terminals, ranging from 3270s to simple ASCII tubes and personal computers. The systems, moreover, are said to be "completely insensitive to the number and type of boxes which are configured"—disk drives can be added or removed while a system is running. File backup is also taken care of automatically, thus freeing further manpower.

"The ability to be responsive to users without incurring tremendous cost penalties is an important facet of the Multics strategy at Ford," the paper states."

"It's unfortunate more people don't realize how good Multics is," says Vin Scarafino, one of the Ford paper's authors. "We've been doing relational database for five years. Our systems handle hundreds of thousands of transactions per eight-hour day. They run themselves and require no babysitting."

Having committed itself so heavily to Multics in the past six years, Ford's Engineering Computer Center has strived lately to pressure Honeywell into budgeting more money for the system. Hence the praises sung in its white paper, which was prepared precisely to help convince other Ford divisions to adopt Multics and thereby enable the company as a whole to obtain greater leverage with Honeywell.

Finally, if programming Multics is easy, trusting it is a cinch. It is simply the most secure general purpose operating system there is, according to the Defense Department's National Computer Security Center, whose charter is to determine such things. Concern for security has permeated Multics since its earliest conception and is achieved through a combination of unique software and hardware facilities. The system takes complete responsibility for data and program security, checking access authorization every time a program instruction is executed. Programmers need not build security functions into their code. By the same token, users can take advantage of Multics' powerful electronic mail and other information-sharing features without fear of messages falling into the wrong hands. Naturally, Multics has found strong acceptance by military organizations in the U.S., Canada, and Europe.

Thus, everyone concerned acknowledges how productive, flexible, and secure Multics can be. "I love that system myself," states Eugene Manno, chief executive of Honeywell's Small Computer and Office Systems Group in Billerica, Mass., which has marketing responsibility for Multics. But with no more Multics hardware on the way, life is getting difficult for users.

"We feel Multics is the system of the future. It has all the advantages of Unix, but it was done right. Multics would be very hard to replace on Unix and especially on IBM mainframes. Honeywell has really hurt us," states MIT's Roach. "We are not pleased, to put it as politely as I can." Other users agree. MIT, one of the top three computer science schools in the world, helped develop Multics in the 1960s and has used it for teaching and research ever since. "We're absolutely furious," says a source at St. John's University in New York, asking to remain anonymous. "I think [Honeywell management] is insane."

CALGARY FINDS IT AWKWARD

"This has put us in an awkward situation," says Norm Barnecut, assistant manager of operating systems at the University of Calgary, in Alberta. The university depends heavily on Multics and, like other sites, cannot com-

THE ROOTS OF MULTICS

General Electric Co., AT&T's Bell Telephone Laboratories, and the Massachusetts Institute of Technology (MIT) all had a hand in developing Multics. They conceived the operating system as the heart of a "computing utility."

Funded largely by the Defense Department's Advanced Research Projects Agency (DARPA), Multics was initially developed under the auspices of MIT's Project MAC.

According to a paper by F.J. Corbato and J.H. Saltzer of MIT and C.T. Clingen of Honeywell, presented to the Spring Joint Computer Conference in 1972, Multics was supposed to form the heart of "a community computer facility" with

1. convenient remote terminal access as the normal mode of system usage;
2. a view of continuous operation analogous to that of the electric power and telephone companies;
3. a wide range of capacity to allow growth or contraction without either system or user reorganization;
4. an internal file system so reliable that users trust their only copy of programs and data to be stored on it;
5. sufficient control of access to allow selective sharing of information;
6. the ability to structure hierarchically both the logical storage of information and the administration of the system;
7. the capability of serving large and small users without inefficiency to either;
8. the ability to support different programming environments and human interfaces within a single system;
9. the flexibility and generality of system organization required for evolution through successive waves of technological improvements and the inevitable growth of user expectations.

Those were ambitious goals in 1964, when the three partners began planning Multics as a cooperative research project. Timesharing was largely untried then and was the focus of much controversy at IBM. As soon as GE won contracts to supply timesharing systems to MIT and Bell Labs, IBM began efforts to design and produce a timesharing version of its recently introduced System/360 line. IBM did not want to lose more of such prestigious accounts.

The result of IBM's efforts was the 360/67, a machine that never lived up to its original technical specifications. That, however, did not stop IBM from taking many orders for the system, which was priced to earn a minuscule profit of only 0.009%. Every order for a model 67, of course, was one less for GE and for Control Data, which also was competing for technical customers interested in timesharing. Indeed, the model 67 effectively stopped GE's computer business from growing, especially as it had its own technical troubles making Multics work. GE threw in the towel in 1970 when it sold its computer business to Honeywell for $234 million.

—J.W.V.

Technically, and perhaps commercially, Multics was always ahead of its time.

fortably plan for future upgrades now. Barnecut is leader of the Multics division of the Honeywell Large Systems Users' Association (HLSUA), but declines to comment in that capacity.

"This will have significant impact on us," states a manager at Ford, also requesting anonymity. Ford is one of the largest, if not the largest, commercial user of Multics. "We certainly are disappointed. Things would be a lot easier if there were alternatives to choose from. We can't afford to go it alone with something this big."

To that end, Ford called a private meeting of Multics users in Detroit to take place April 5, a day before a semiannual meeting of HLSUA was to begin in that city. Ford's invitation to other users of Multics to help "counter this abandonment" states: "As you are aware, individual site and collective HLSUA action has done little to convince Honeywell that they are causing a great deal of grief and harm to their Multics customer base." Several competing vendors, including Control Data, Digital Equipment, Amdahl, and AT&T, were invited to the meeting to see whether or not they are interested in supplying future Multics machines.

Ford's actions and the criticisms of Honeywell follow years of increasing uncertainty for Multics users. Honeywell's actions, they contend, cast doubt on its credibility as a computer vendor. They denounce the killing of Multics as a sign of Honeywell's "apparent technical incompetence," and argue that had Multics been marketed wholeheartedly from the start, the current situation would probably have been avoided.

Technically, and perhaps commercially, Multics was always ahead of its time. It was conceived in the mid-1960s, when industry leader IBM had decided timesharing offered minimal commercial potential (see "The Roots of Multics"). Honeywell gained control of the system in 1970 when it bought General Electric's unprofitable computer operations. At that point, the software was still only a research project at MIT, not a commercial product. Honeywell failed to recognize Multics' commercial potential and began a pattern of favoring the widely installed GCOS system, also acquired from GE. Some farsighted managers saved Multics from the scrap heap, and it was kept going as an internal software development system until it was finally brought to market for the first time in 1973.

"Honeywell never resolved the contention between those two competing mainframe products," recalls Roger R. Schell, who worked on Multics in the early 1970s and is now vice president of engineering at Gemini Computers Inc., Carmel, Calif. "They had an opportunity to integrate the two systems at the software and hardware level but they did it only at the hardware level."

MULTICS LOST OUT TO GCOS

Schell and others present at the time recall a tug-of-war between the Multics and GCOS factions at Honeywell. GCOS won, it is believed, simply because it was generating substantial revenues. The Multics faction could boast only of technical superiority, but that did not get much of a hearing at the company's upper levels, where an "IBM mentality" ruled: GCOS more closely resembled IBM's OS than did Multics and seemed a safer bet with users.

"They were afraid to look at Multics too closely," says Schell, "because they might find GCOS was second class. They tried to aim the two systems at different markets but that was not the right thing to do."

Multics, avid users recollect, was clearly superior to every timesharing system against which it had to compete, particularly 1970s-vintage TSO running on batch-oriented IBM 370s. They believe that Multics would probably have scored better with new customers if a more knowledgeable, better motivated sales force had been handling it.

From the beginning, Honeywell "never understood how to market [Multics]," says a former Multics developer who consults now. "I think Honeywell should have gotten out of it years ago." A folklore has arisen about Multics and how poorly it was marketed. One anecdote has Honeywell buying a Wang word processor to prepare a bid that proposed Multics as the perfect text system.

True or not, there are many stories about how hard it was to find a Honeywell salesman with enough knowledge of Multics to sell it. Often, it is said, Multics failed to impress potential customers because its structure was so radically different from IBM systems. For instance, Multics has no RUN command to start program execution, nor does it have a link editor as such.

"Honeywell has never marketed Multics," HLSUA wrote to Honeywell's chief computer executive, Dr. James J. Renier, vice chairman of the company, in a letter last August. "We've tried to communicate with Honeywell Senior [sic] management, but don't believe you're listening."

Users blame Multics' impending demise entirely on certain managers, Renier among them. To use some of the gentler adjectives in circulation, they are described as "totally out to lunch," "Neanderthalic," and "schizophrenic." Even some of Honeywell's own Multics staff, it is said, privately denounce their management: resident systems engineers at one East Coast Multics site have jokingly "wished" with their customers that the company's corporate jet would crash. Renier declined to be interviewed for this story. Honeywell's Manno, who is responsible for selling Multics, declines to comment on users' criticisms.

"There is a case to be made that Honeywell would like nothing better than to have Multics die quietly and then slip out of the computer business as profitably as possible," says one veteran Multics hand, who requested anonymity.

The increasing tension between the Multics crowd and Honeywell in the past year has centered on the lack of any future Multics hardware. During 1984 and early 1985, Renier and Manno dropped in at several large U.S. Multics sites to say that the company was "committed" to the product, Manno confirms.

Users say that Renier promised Honeywell would undertake a major effort and investment that would produce a larger Multics mainframe employing advanced technology and entail a major marketing thrust. It was to be, Renier is said to have told them, "a multiyear commitment."

"He told us Multics would be a cornerstone for the future," recalls one Department of Defense user. "We feel they [Honeywell] have been unfaithful business partners after that."

MARKET NOT BROAD ENOUGH

In April 1985, the company announced that it had canceled plans to build a new 36-bit Multics machine, the so-called Flower processor, because it had failed to identify a broad enough potential market for the hardware. Flower was to have been built from custom VLSI logic, but that technology requires high volumes to be profitable—and no volume, no production was Honeywell's justification. Instead, the firm said it would make certain features of Multics available on the forthcoming line of 32-bit superminicomputers, named HRX and due for introduction this year. The unspecified Multics features are not slated to show up until 1988, however.

Users said they couldn't wait that long. They needed Flower, or something like it, immediately. They charge Honeywell failed to find a broad market for Flower because it has continued to misunderstand Multics. Multics has no place being marketed by Eugene Manno's Small Com-

Multics may ultimately be perceived as the classic its devoted users claim it to be.

puter and Office Systems group, they say—it is not small and it is not an office system. Manno defends his responsibility on the grounds that Multics at one time was supposed to serve as a hub for networked office systems, but that plan never panned out.

"I put together a blue-ribbon study team for Multics, and I started and encouraged the Flower project," Manno says. "But when it was completed, we found Flower had no appeal outside the existing Multics user base. Its footprint was too big and it had a noncontemporary architecture. There was no way to market it. So we put it on the shelf."

Multics users quickly rallied forces and appealed to their user organization, HLSUA, for help. After a list of the major factors that they thought contributed to Multics' "excellent market potential," the users noted in an April 17, 1985 letter to the HLSUA board: "The consensus of the current Multics customers is that, although there are exceptions, historically and currently, the Honeywell sales approach concerning Multics has not only been nonexistent but at times counterproductive. . . . This can only be an indication that Multics is an excellent product which to date has sold itself. . . . A renewed commitment to aggressively market Multics was announced just last year. It appears that discussions to discontinue the product are extremely premature. . . . This has become an issue which causes us to ask the question of whether or not Honeywell is serious about the computer business."

Late August, HLSUA fired off its letter to Renier stating that its Multics members found Honeywell's plans "unanimously unacceptable." What they wanted, HLSUA told Renier, was a "Multics solution that retains the current Multics product . . . and includes both hardware and software development, not merely support." HLSUA suggested Honeywell consider reviving the Flower program to build a modern, 36-bit Multics processor; or consider engaging another manufacturer to build such a machine; or "port" Multics "in its entirety" to the 32-bit line of superminis Honeywell was designing. The letter to Renier noted that Multics users had "extreme interest in pursuing alternate vendors" in the event the company decided not to change its mind regarding the operating system's future.

"We perceive you are still willing to write the Multics community off," an attached list of grievances stated. "This is a delaying strategy instead of a solution." It noted, too, that for Multics users to convert their software to run on non-Multics machines might cost as much as a billion dollars.

Honeywell responded in October, saying it had concluded that neither further 36-bit Multics machines nor porting Multics to a 32-bit machine were "viable alternative[s]." The possibility of having another party "pursue a different course with Multics" was still being explored.

TRIED TO LICENSE FLOWER

Previous attempts to license Multics from Honeywell had been made, but nothing much came of them. Rome Air Development Center tried a few years ago, according to sources there. The proposal made last fall by Olin Sibert of Arlington, Mass., was more substantive. Acquainted with Multics for the past 11 years as a user and consultant, Sibert pulled together a consortium of investors that sought to license the design of Honeywell's Flower processor. Sibert says his group wanted to produce a less sophisticated but nevertheless useful and marketable version of Flower that would employ low-power Schottky logic instead of the higher-cost, custom VLSI that Honeywell had envisioned.

"I proposed taking conventional, low-risk hardware and building a processor that would provide a substantial improvement over Honeywell's DPS 8," Sibert says. The DPS 8 systems, he says, are "physically huge. Several customers are unable to expand their systems because of problems with floor space and the demand for more power."

Sibert says that Honeywell stopped negotiations with him after he discussed his proposed machine with users in too much detail before a deal had actually been signed. "I've been in the doghouse with them ever since," Sibert says. Manno says Honeywell called off negotiations because it was unsure of Sibert's ability to raise enough money to be successful.

Several users, including hard-pressed Ford Motor Co., say Sibert's machine is still worth considering. Manno says Honeywell has shown Multics to "eight different vendors" as potential licensees, but found "they don't get very close."

Of all machines on the market, Control Data's mainframes seem most compatible with Multics, users say, but that company's financial troubles do not bode well for a takeover of the operating system, even if Honeywell agreed to such a deal.

As of late March, the Multics clan said it was still hoping to salvage Multics intact. The Ford meeting held some promise—the unlikely but nevertheless appealing idea of Ford simply buying Multics from Honeywell was still being kicked about. In France, where Multics has done particularly well, there was talk of a class-action user suit against Groupe Bull to pressure it into taking over Multics development.

"It will be a real pity if all the great work in Multics gets thrown away and people have to start from scratch," adds John M. Strayhorn, president of Renaissance Computing Inc., a Cambridge, Mass., software house that markets the widely regarded Consistent System program for Multics.

Perhaps uniquely in this infant industry of perpetual change, Multics may ultimately be perceived as the classic its devoted users claim it to be. It is, they say, one of those graceful machines like the MG-TD road racer or Leica camera that is mimicked but can never be matched. But, as an object that is being constantly enhanced even as its essential form remains the same, Multics is more, users claim: its concept is so artful, its algorithms so refined, that it has survived in spite of those who misunderstand it. ◉

"The joke's on my wife — she always said I was hopelessly middle-class."

CARTOON BY REX F. MAY