"LET US NOW PRAISE FAMOUS HACKERS" was the headline in the Time magazine story of the Hackers' Conference. Famous, infamous, and unknown, they praised each other with nightlong attention and exuberance. Left to right here: Mike Coffey, Steve Capps, John Draper, Andy Hertzfeld, and Dick Heiser, who had the final word: "Keep designing."

HACKERS

CONFERENCE • 1984

# "KEEP DESIGNING"

## HOW THE INFORMATION ECONOMY IS BEING CREATED AND SHAPED BY THE HACKER ETHIC

*Edited by Stewart Brand* *Photographs by Matt Herron*

*I think hackers — dedicated, innovative, irreverent computer programmers — are the most interesting and effective body of intellectuals since the framers of the U.S. Constitution.*

*No other group that I know of has set out to liberate a technology and succeeded. They not only did so against the active disinterest of corporate America, their success forced corporate America to adopt their style in the end. In reorganizing the Information Age around the individual, via personal computers, the hackers may well have saved the American economy. High tech is now something that mass consumers do, rather than just have done to them, and that's a hot item in the world. In 1983 America had 70 percent of the $18 billion world software market, and growing.*

*The quietest of all the '60s sub-subcultures has emerged as the most innovative and most powerful — and most suspicious of power.*

*Some of the shyer people you'll ever meet, hackers are also some of the funniest. The standard memory of the Hackers' Conference is of three days and two long nights of nonstop hilarity.*

*These supposed lone wolves, proud artistes, in fact collaborate with glee. Though famous as an all-male tribe, they have zero separatist jokes in their style; they comfortably welcomed the four female hackers (of 125 total) at the conference, and a couple of romances blossomed.*

*Like the prose of poets, there is impressive economy in the conversation of hackers, whose lifework is compressing code, after all. What follows is an only-mildly-edited transcript of one morning discussion on The Future of the Hacker Ethic, moderated by Steven Levy. Thirty-six voices are heard. Some are millionaires, some are quite poor. In how they treat each other, you cannot tell the difference. —SB*

*Some of the most high-powered pioneers in the computer business were gathered to reassess their origins. In a now intensely commercial business, they found they still were wanting to keep the faith in what they variously called the hacker drive, the hacker instinct, the Hacker Ethic.*

*In a new book called* **Hackers: Heroes of the Computer Revolution,** *which was the inspiration for the conference, tenets of the Hacker Ethic are stated as: 1) Access to computers — and anything which might teach you something about the way the world works — should be unlimited and total. Always yield to the Hands-On Imperative! 2) All information should be free. 3) Mistrust Authority — Promote Decentralization. 4) Hackers should be judged by their hacking, not bogus criteria such as degrees, age, race, or position. 5) You can create art and beauty on a computer. 6) Computers can change your life for the better.* *—SB*

## DISCUSSIONS FROM THE HACKERS' CONFERENCE, NOVEMBER 1984

STEVEN LEVY *(author of* **Hackers***):* The Hacker Ethic, as I think all of you know, isn't something which back at MIT in the early '60s people would raise their hand and say, "I vow to follow the hacker ethic." It's a term I used to describe what I felt was a shared philosophy, not only of the hackers at MIT, but the people in the Homebrew Club who designed the first small computers in the mid-'70s, and some of the younger people who started hacking with those small computers later on.

BILL BURNS *(Homebrew-era hobbyist):* Steve, can a person be a hacker without being the kind of superstar or wizard that you're talking about in the book? Can somebody be a low-level hacker just because he wants to have fun and an intellectual curiosity about the computer? Even though maybe he's not very good as a coder?

LEVY: One issue that I found at MIT was that some people were complaining for that very reason — that you had to be a "winner," you had to be really good to be considered a hacker.

BRUCE WEBSTER *(co-author of SUNDOG, a great capitalists-in-space game):* One of the ironies in that is that "hacker" originally denoted someone who wasn't very good. It was someone who was not skilled professionally but tried to make up in volume what they couldn't produce in quality. *(laughter)* Or at least he was using a shotgun rather than a high-powered rifle.

RICHARD STALLMAN *(MIT system hacker, author of EMACS):* You're always gonna find that if there's a community of real wizards they're gonna lose patience with the people who aren't. That doesn't mean that they can't be real hackers.

VOICE: The question is, "Can you hack in BASIC?"

CHORUS: Nooooo!

ROBERT WOODHEAD *(co-author of WIZARDRY, the classic role-playing adventure game):* Only if you're very *very* good can you hack in BASIC. *(laughter, applause)*

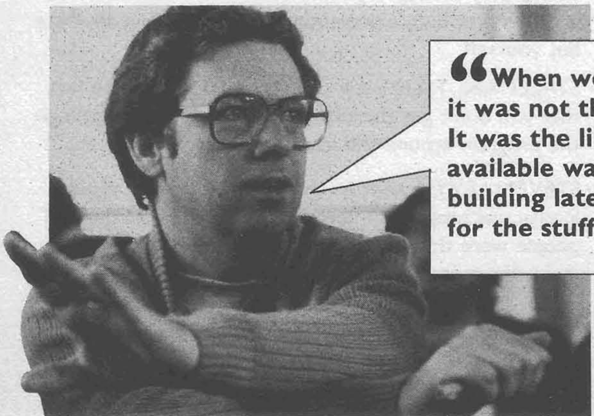
BRIAN HARVEY *(former MIT and Atari hacker, now working with kids):* The term "hack" at MIT predates computer hacking. The way it started out, there were two kinds of people. There were "tools," who were the ones who went to all their classes and when they weren't in class they were in the library. And then there were "hackers," who never went to class and slept all day and did something or other all night. Before it was computers it was model railroads, or telephones, or movies, or Chinese food, or anything. Hacking started out as not something technical (although it tended to be technical, because this is MIT we're talking about), but a sort of approach to what's important in life. It really means being a hobbyist and taking your hobby seriously. If programming, for example, is something that you do on Sunday afternoons and the rest of the time you don't think about it, then you're not a hacker. But you don't necessarily have to be a star to be a hacker.

Now, if you're at the MIT A.I. (Artificial Intelligence) Lab, at least if you were there when I was there, you *did* have to be a star in order not to get dumped on a lot. And that was the problem. It was something that I hated very much.

DENNIS BROTHERS *(author of MACTEP, the first telecommunications program for the Macintosh):* It should be pointed out that, at least by the time I got there, '64 or so, "hack" meant "a prank," plain and simple, and the better the prank the better the hack. Things like the big black moon at the Harvard-Yale game was the ultimate hack.

PHIL AGRE *(MIT A.I. Lab):* These days at the A.I. Lab, the word "hack" is very, very diffuse. It is one of the very large number of content-free generic words, like "frob" and "the right thing," that fill the hacker's dictionary. I get the impression from the olden days that it once meant something more focused, but I'll be damned if I can figure out what it was.

STEVEN LEVY: Well, without focusing a whole lot on the word, I think there's pretty much an agreement here that there's a resentment of using the word totally to mean breaking into computer systems, and we are talking about it in a broader sense. How much of what we see now in programming has that same kind of



**"When we were hacking around in the mid-'60s at Harvard, it was not the engineering students who were the hackers. It was the liberal arts majors whose only computer time available was if they gummed up the locks and snuck into the building late at night because they weren't allowed to sign up for the stuff."** *—DOUG CARLSTON, founder and president of Broderbund, publisher of computer games*

*Left, in winged cap, Ted Nelson (*Computer Lib*) moderates a session on "Tools for Hackers." Above, Howard Pearlmutter "flaming" (expressing strong and extended opinion).*

devotion, non-dilettantism, that we saw in the days when people had to stay up all night just to get computer time?

DOUG CARLSTON *(founder and president of Broderbund, publisher of computer games):* May I protest just a little bit? When we were hacking around in the mid-'60s at Harvard, it was not the engineering students who were the hackers. It was the liberal arts majors whose only computer time available was if they gummed up the locks and snuck into the building late at night because they weren't allowed to sign up for the stuff. You did everything by trial and error, because we didn't have any courses, we didn't have access to anything other than manuals, and as far as I'm aware the whole group of midnight programmers there were people who didn't have any real functional use for what they were doing at all. So we called ourselves "hackers."

BRUCE BAUMGART *(early Stanford A.I. hacker):* I was at Harvard in the same years when I found the PDP-1 at the Cambridge electron accelerator and to stay up all night with it was just incredible. You could roll in at 9 P.M. when the physicists had left and you could stay there till 9 A.M. when they rolled back in. Do it night after night. I made it to classes but I slept through them.

STEVE WITHAM *(Xanadu, which is a scheme for a worldwide database and writing system founded by Ted Nelson):* It's not so much a hacker ethic as a hacker instinct. It's sort of like the baby ducks when they see their first moving object. *(laughter)*

RICHARD STALLMAN: You see your first computer language and you think, "This language is perfect." *(laughter)*

MARK MILLER *(Xanadu):* The computer itself is really the first moving object in some sense that any of us have seen. I think that what creates the hacker drive (I won't call it a hacker ethic, and I want to argue about that) is that there's a sense, "There's something terribly important here." It goes beyond the effect that this thing can have on the world and what I can do with it and all that. "There's something essential here to understand and I don't know what it is yet." I still don't know what it is.

STEVE WOZNIAK *(designer of the Apple computer, cofounder of Apple Computer, Inc.):* I think the hacker drive represents the children in us. Children love to discover, explore, create something a little beyond what they could before. In school you have the courses that teach you the problem and the solution, whereas the hackers tended to be just bright enough to take the little starting points, the mathematical tools, and build up a solution of their own, and they could discover the optimum solution of the day. The hacker motivation is what's different. They were intrinsically motivated; the challenge of solving the puzzle was the only reward. The rewards were in their head. It was like a hobby, whereas in the outside world they would have a job, careers, advancements, salaries — extrinsic rewards.

MARK MILLER: The reason I argue against the "hacker ethic": I think that Steve Levy's book was wonderful and I enjoyed it a lot, but I very much resented the way it, I think, tried to shoehorn in this idea that hackers as a group were necessarily against the idea of intellectual property. I considered myself a hacker in school, I consider myself a hacker now, and I've always thought that the idea of intellectual property was a good one.

RICHARD STALLMAN: There is definitely a tendency for hackers to not put up with someone who wants to deliberately obstruct them from doing something that's a fun hack. If somebody says, "It's useful for my purposes to prevent people from doing this in-itself-innocent activity, such as prevent people from logging in if I haven't given them accounts, or prevent people from running this program just because I'll get less money if they can run this program," . . .

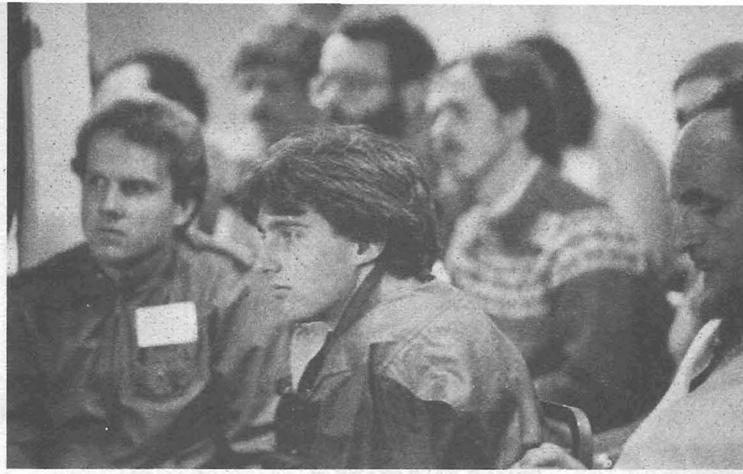
VOICE: And use lots of undocumented entry points.

STALLMAN: If the person doesn't see a good reason why he shouldn't run that program or why he shouldn't use that computer, if he's a hacker, he'll tend to view the bureaucracy that stops him as a challenge rather than as an authority that he must respect.

BILL BURNS: The drive to do it is so strong that it sweeps other things aside. I think this is one of the big differences between the people that do their hacking on computers that cost a lot and are owned by other people, and the people that do their hacking on micros where they own it. If you own the micro there's no us and them, nobody's preventing you from doing anything but yourself.

STALLMAN: There's still copy-protection, and the fact that you don't get the source [codes]; you can't change the program around and learn something.

STEVEN LEVY: I want to answer Mark's point about intellectual property. I never meant to say that the MIT people were these fantastic people who didn't want to make any money ever. The fact was, for example, in



*Above, Richard Stallman (MIT): "You see your first computer language, and you think, 'This language is perfect!'" Far right, Les Earnest (Imagen): "There are very few team hacks that one can think of that went anywhere."*

'61, when Steve Russell wrote SPACEWAR *[the earliest and greatest computer game for 12 years]* as a hack and some people in the room helped improve it, the improvements came because it was an open program. Of course, Steve couldn't possibly have made any money by releasing SPACEWAR as a product, since I think there were only fifty PDP-Is in total made. Because he had that advantage that no one was tempting him, it was very natural to just leave the program in the drawer, let anyone look at the code, improve it, and what happened was you got a much better product from it being a universal property. In some more "serious" things like assemblers and compilers and all sorts of utility programs, the same system benefited everyone there. I think things happened that wouldn't have happened if programs were sequestered away and kept proprietary.

UNIDENTIFIED HACKER: There's one community in which this system does work, and that's academe, in particular the community that MIT is. In academia you're valued by how much you publish. The whole point is to discover something and at the end give it away. And if I could get a reasonable full professorship writing software and giving it away, I'd be very happy to do that.

What I'm doing is something like science but different from science, because in science I'm pushing the boundaries discovering new things. But only in computers do those things that I discover wrap around and increase my ability to discover the next thing. Computers have this nice feedback, positive feedback, that everything I do on my computer makes it better for me doing more things on my computer. No other field works that way.

VOICE: Organic chemistry works that way. All fields work that way.

BRUCE BAUMGART: I think we've forgotten something there, which is the bad nights at the lab, when the hackers stepped on each other's toes, when you were trying to get a paper done and somebody was hacking the text editor. You were trying to take a television picture, and somebody was running music using up all the disk space. There was anarchy. The big dogs would survive. You would go home, your stuff undone, because somebody bigger than you and more powerful than you and knew more codes, whatever, had stepped on you, or your disks or your pictures or something. Didn't you have bad times? Or were you always the biggest dog on the machine?

RICHARD STALLMAN: I always tried to oppose having it be a society of dog eat dog. I never tried to eat the dogs that were smaller than me. Whenever a person tried to act toward me as if I were above him, I'd always say, "I'm not above you; do what you think you should do; you shouldn't get orders from me." And if somebody thought he was above me, I would say, "You can't give me orders. See if you can get me fired; I want to do what I want."

BRIAN HARVEY: I think we're trying much too hard for a sort of unanimity here that doesn't exist about what all of us hackers are like. For example, if you want to bring up the word "ethics" — I felt very uncomfortable last night with a couple of people who got up and talked about how they made their living by stealing from the telephone company. I think it's one thing to be a high school kid wanting to show off that you're capable of making a phone call without paying for it, and it's something else to be an adult being in the career of encouraging people to be thieves.

STEVE WOZNIAK: I'd like to discuss the telephone topic from a hacker perspective, and it applies to software piracy. There are some people that actually have money and are ethical. Back then we went out and treated telephone blue boxing and the like as a fun exploration of the phone system. How could we make every call in the world, in every nook and cranny and all that, but I'll tell you, my phone bill as a college student at Berkeley was very high because I paid for all the calls I would have paid for anyway. I only used the phone system to explore the network. Some pirates copy software and they'll copy everything and put it in their collection, but if they find something that they do like and decide it's a good one, they'll go out and buy it because the producer deserves the money.

BURRELL SMITH *(designer of the Macintosh)*: I think one of the common threads of hacking is that all of us want a very pure model of what we're working on. Nowadays we're all very complex, we have stock options, salaries, and careers and stuff. Back then it was the joy of being absorbed, being intoxicated by being able to solve this problem. You would be able to take the entire world with its horrible problems and boil it down to a bunch of microchips or whatever we were hacking.

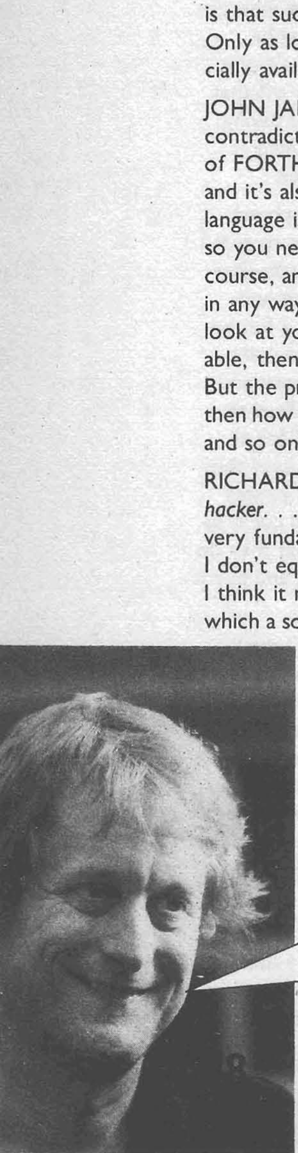
I think another aspect of that is that hackers can do almost anything and be a hacker. You can be a hacker carpenter. It's not necessarily high tech. I think it has to do with craftsmanship and caring about what you're doing. The joy of seeing your stuff work is the excitement.

STEVEN LEVY: Yeah, but aren't there contradictions you have to deal with when those stock options and things like that get in the way? Homebrew had a period before there was a whole lot of money, when people would come in and say, "Here's the plans to this computer we're coming out with." Then there started to be secrets kept. How do you keep things going forward as much as possible when you have to keep those secrets, when you have allegiance to your company and its proprietary stuff?

BRUCE BAUMGART: You just graduated from the academic to the commercial. There's many worlds, and I think the worlds overlap.

RICHARD STALLMAN: The question is, does one of them eat up the other so that it goes away? That's what seems to happen.

TED NELSON *(author of* Computer Lib/Dream Machines, *founder of Xanadu):* A perspective that hasn't been mentioned is that in times like the Homebrew Club, people had jobs. As Thomas Jefferson said, "I make war so that my grandchildren can study philosophy." The person who is studying philosophy is at the top of a food chain. *(laughter, applause)* The problem when the philosophers find they can sell philosophy is that suddenly it's the bottom of a food chain again. Only as long as it wasn't something that was commercially available could it have this pure aspect.



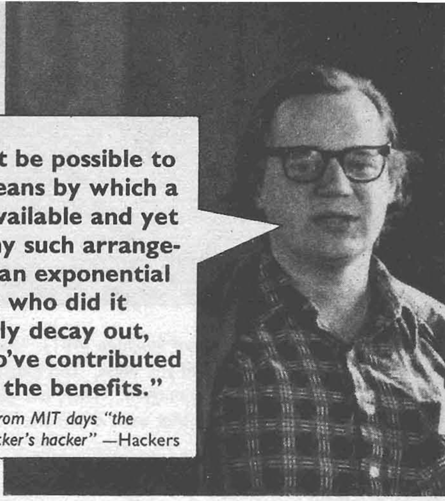
**"The person who is studying philosophy is at the top of a food chain. The problem when the philosophers find they can sell philosophy is that suddenly it's the bottom of a food chain again. Only as long as it wasn't commercially available could it have this pure aspect."**

*—TED NELSON, author of* Computer Lib/Dream Machines; *founder of Xanadu*

JOHN JAMES *(FORTH hacker):* There's a certain kind of contradiction that we're still dealing with in the world of FORTH, where the public domain is the soul of it and it's also the curse. The advantage of a programming language is that you can do anything you want to do, so you need complete access to the source code, of course, and then you need to be able to use the products in any way you want without having to let somebody look at your books in all future time. If that's not available, then the advantages of FORTH really aren't there. But the problem is that if everything is public domain, then how do you support elaborate systems development and so on? That's what we really haven't dealt with.

RICHARD GREENBLATT *(from MIT days "the archetypal hacker. . . the hacker's hacker"* **—Hackers***):* I think it's very fundamental that source codes be made available. I don't equate that with giving them away necessarily. I think it might be possible to work out some means by which a source code was available and yet it was licensed, on a basis that didn't involve a great deal of bureaucrat overhead to the proceedings. If that could be done then you would get the best of both worlds. The people who had written something originally would have the benefit of some royalties; they would also have somewhere in there "copyright so-and-so," and it would be recorded that they were responsible for a particular piece of code.

Having thought about this a lot, I've come up with only a few ideas to try to make it practical. One of them I think is that any such arrangement should have an exponential tailoff. In the first year the royalties should be such and such percent; after another year the royalty goes down one-half of what it was previous, or something like that — so that the royalty pie doesn't just get bigger and bigger, but the people who did it originally eventually decay out, and the people who've contributed more recently get the benefits.



**"I think it might be possible to work out some means by which a source code was available and yet it was licensed. Any such arrangement should have an exponential tailoff. The people who did it originally eventually decay out, and the people who've contributed more recently get the benefits."**

*—RICHARD GREENBLATT, from MIT days "the archetypal hacker . . . the hacker's hacker" —*Hackers

STEVE WOZNIAK: Hackers frequently want to look at code, like operating systems, listings, and the like, to learn how it was done before them. Source should be made available reasonably to those sort of people. Not to copy, not to sell, but to explore and learn from and extend.

ROBERT WOODHEAD: Well, as a dedicated capitalist exploiter of the masses and running dog lackey of the bourgeois, I find that the software that I write usually falls into two different categories. There are finished products like WIZARDRY that I sell and make a living on, and then there are the tools that I wrote to build those products. The tools I will give away to anybody. But the product, that's my soul in that product. I don't want anyone fooling with that. I don't want anyone hacking into that product and changing it, because then it won't be mine. It's like somebody looking at a painting and saying, 'Well, I don't like that color over there, so I'll just take a can of paint and change it."

JERRY POURNELLE *(science fiction writer, columnist in* Byte *magazine):* You never had to deal with editors. *(laughter)*

WOODHEAD: Yes I do. I tell 'em to go to hell. On the other hand, if somebody sees something I did and says to me, "How did you do that?" I'll tell 'em in a minute. I'll give them all the information they need so that they can go out and do something better, because what I want to see is really great stuff. That's why all the tools I've developed when I've been working on the Lisa, I regularly send them off to Apple so that they can get them out there, because I know they're gonna help somebody. Then something really great's gonna come out and take away all the market sales of my product. Then I'm gonna have to go out and write a better one.

BOB WALLACE *(author and distributor of PC-WRITE[1], an outstanding word processing program for IBM PCs and compatibles):* We give away source with our product, and we haven't found it to be a problem. We do what we call 'Shareware." We give away PC-WRITE, and it seems to be supporting us, you know.

When I started, I wanted to do a product and I wanted it to be self-supporting. I didn't want to do it for another company and have somebody else have control over it. I wanted to have control over it *and* I wanted to make a living. Not having a lot of money for advertising, I figured the way to distribute it was, you know, word of disk. Diskettes are a new medium that I don't think people have realized how easy they are to copy and what that means, but it gives us a distribution channel.

It's very hard to get shelf space in stores. But most people choose their software based on recommendations by other people — 40 percent, I think. Next comes product reviews and next comes advertising. With PC-WRITE, people can not only recommend it but they can give it to somebody. People want to feel like they can use the software for a month or two and see, "Is this my software?" How many people here have bought a $500 package and discovered, "Well, it isn't quite what I need," and you're out $500?

STEVEN LEVY: You do get royalties?

BOB WALLACE: Yeah, people do send me money. People after they're using it want to feel safe, they want to feel like there's support, they want to feel respectable and part of a larger process, and they want to support companies they like. So they send us money. Support includes a newsletter and updates and phone support and the source code. We've done fairly well. We've sold 6,400 $10 diskettes, and about 1,700 people then registered for $75. Then we also sold some on an OEM basis *[Original Equipment Manufacturer, where a hardware maker or distributor includes software with the machine purchase]*, a couple thousand that way, because once you're out, and people have heard of you, then you can start working quantity deals where people'll buy your source and modify it and send you royalties.

**"Giving software away is a lot of fun. You get great letters and great phone calls, people are very appreciative, and they give you some great ideas. At the same time we'll gross about $225,000 this year."**

*—BOB WALLACE, author and distributor of PC-WRITE*

LEVY: Was all that solely a marketing decision?

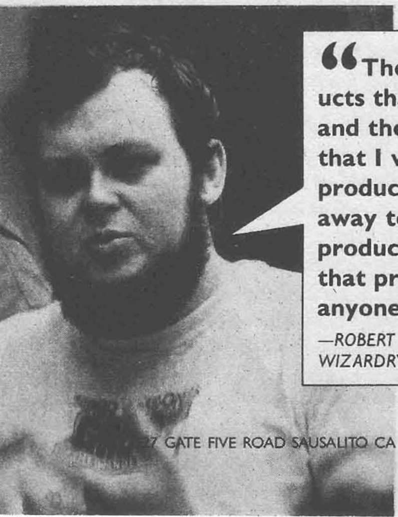
WALLACE: It was a way to do what I wanted to do without getting involved either in another company or with venture capital. And giving software away is a lot of fun. You get great letters and great phone calls, people are very appreciative, and they give you some great ideas. At the same time we'll gross about $225,000 this year. It's supporting two of us; we're adding a third person. So you can start a small company that way. I don't know how far we can get, I don't know how many people would send in voluntary registration money to Microsoft or something like that.

STEVE WOZNIAK: In a company sometimes a product gets developed and the company decides it doesn't fit a market, it won't sell. In a case like that the company should be very free to quickly give it to the engineer, legal release: "It's yours, take it out and start your own company." But sometimes the companies, because they own the product, will squash it and say, "You cannot have it, even though we're not gonna put it out, and nobody else in the world's gonna get it." That's a hiding of information, and that is wrong.

STEWART BRAND *(author of "Spacewar: Fanatic Life and Symbolic Death Among the Computer Bums," 1972):* It seems like there's a couple of interesting paradoxes that we're working here. That's why I'm especially interested in what Bob Wallace has done with PC-WRITE and what Andrew Fluegelman did before that with PC-TALK. On the one hand information wants to be expensive, because it's so valuable. The right information in the right place just changes your life. On the other hand, information wants to be free, because the cost of getting it out is getting lower and lower all the time. So you have these two fighting against each other.

WOZNIAK: Information should be free but your time should not.

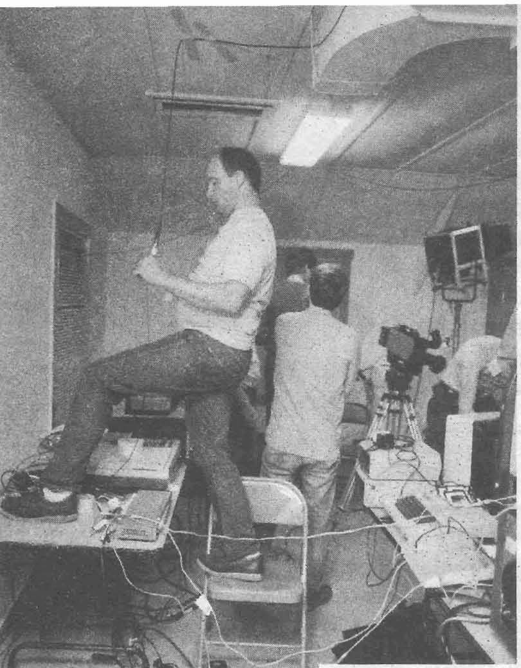
BRAND: But then, at what point of amplification is your time being so well rewarded that it's getting strange or so under-rewarded that it's strange? There's problems there with the market. ►

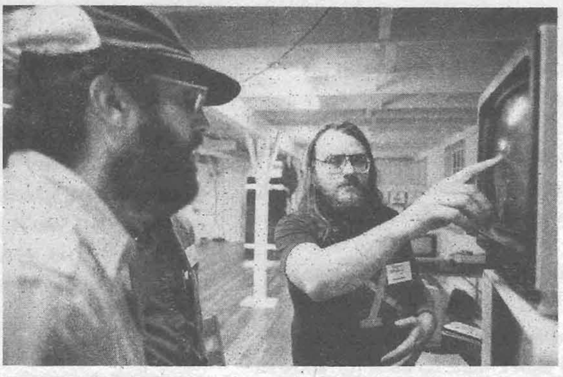


**"There are finished products that I make a living on, and then there are the tools that I wrote to build those products. The tools I will give away to anybody. But the product, that's my soul in that product. I don't want anyone fooling with that."**

*—ROBERT WOODHEAD, co-author of WIZARDRY, classic adventure game*

[1] PC-WRITE is freely copyable and you are encouraged to give it to friends. You can get it by mail for $10 postpaid from Bob Wallace, Quicksoft, 219 First N. #224, Seattle, WA 98109. If after using it you decide to register your program, send Bob $75, and you'll get phone support, a bound manual, and the next version of the program. Also, if anybody to whom you give copies of your PC-WRITE decides to register, you are sent $25 of their $75.
A pyramid scheme without losers. —S.B.





Left, Roger Gregory challenging disbelief in Xanadu, a database-to-be the size of the world.

Below, catered brunch outside the old Army chapel, a touch of style to end the bare-bones weekend in the barracks of Fort Cronkhite, now part of the National Park System near Sausalito, California.

Above, Jim Stockford setting up for "Graphics and Music Evening."

Right, Dave Caulkins (Packet Technologies) drawing a distinction about who influenced whom on the Hackers' Genealogy Chart.

*Organizing the Hackers' Conference was like some of the early hacking at MIT, so collaborative and rapid you couldn't keep track of who did what. It practically fell together around the strength of character and curiosity of the participants.*

*Kevin Kelly imagined such a conference after reading* Hackers. *I instigated the thing, and Patty Phelan was loaned half-time by John Brockman Associates to organize it in two months. Designing the conference itself were four hackers: Lee Felsenstein, famed as the master of ceremonies of the Homebrew Computer Club at its height, designer of the Osborne I and of Community Memory; Bill Budge, author of* PINBALL CONSTRUCTION SET; *Andy Hertzfeld, prominent on the Macintosh Development Team, co-designer of the Thunderscanner; and Doug Carlston, founder and president of Broderbund Software, Inc. Steven Levy, along with Whole Earth's Art Kleiner, Matthew McClure, and Kevin Kelly, played essential roles in continuity and follow-through. Office Manager Lyn Gray handled relations with the site, Yosemite Institute at Fort Cronkhite, where she used to work.*

*It was set up as an invitational conference, no featured speakers, minimal budget; all participants paid the same $90 for food and lodging and conference no matter how much or little they used. With $5,000 donated by Doubleday (publishers of* Hackers *and* The Whole Earth Software Catalog) *the event broke smoothly even at a modest total of $16,500.*

*Knowing we had facilities for 150, over 400 hackers were invited in three waves. That became a saga of its own — identifying the right 400, getting current addresses and phones of a slippery crowd, getting them to respond (many are beleaguered; many ignore mail and phone). But once they were on the scene, they were the world's easiest group to work with. If anything went wrong, 1) they didn't care, 2) they could fix it. Staff, volunteers, participants, and press (20 knowledgeable computer reporters were invited) blended into one energetic population. As* PC Week *headlined the following week, "*HACKERS FIND WONDER, EACH OTHER AT CONFERENCE.*"*

*Thanks to a $5,000 donation by Steve Wozniak, the entire amazing weekend was videotaped — making this article possible, as well as a segment on PBS's* High Tech Times. *Videoist Fabrice Florin (624 Cabrillo, San Francisco, CA 94118; 415/751-8888) is seeking modest financial assistance to edit together a half-hour broadcast version of the historic occasion. —SB*

Then there's another paradox which is especially visible here. This conference is primarily programmers, almost no one who is primarily marketing. In the last year or so the marketing people drove the business, and they're having a tough year. *(laughter)* And nobody's really sorry about that. There's an opportunity now for the programmers, the creators, the fountainhead to reestablish where the initiation of this stuff comes from. Where it begins.

WOZNIAK: You get a lot of problems when you get engineers who are interested just in the technical solution, the right solution. It's got an incredible value to them because it was an incredible discovery, it took a lot of work to find it, and they pay no attention to marketing considerations. Somebody has to *use* this thing eventually. It has to make sense as a product. Sometimes engineers are in control and cause the most disastrous consequences for the companies in this business, because they did not act as one person with marketing.

BRAND: One of the problems with all that brilliant research at Xerox PARC — which was wasted at Xerox and later at Apple turned into the Macintosh — is that they never got to cycle their stuff through product. They never got to really deal with customers the way Wallace does or Fluegelman does, where they have a direct pipe between themselves and the people who are using their stuff. And since the Shareware guys are not fighting their own inventory (because they don't have to have any), they can respond with new improvements, new versions all the time. What they're doing strikes me as the best solution so far to these paradoxes. One of the things I'd like to see shared here is the economics of how to be in business for yourself or in cahoots with other designers, and have the marketing guys working for you.

WOZNIAK: Frequently you have the engineering here and marketing there, partitioned. It's much better when the engineers have a lot of marketing content and the marketing people have a lot of engineering content. It's much more motivating and more productive.

TERRY NIKSCH *(Homebrew hacker)*: Yeah, but I think you're almost getting into a definition there. I think a hacker works to please himself first and to impress his peers, but as soon as you go for institutional approval, which includes the institution of the marketplace, I don't think you're hacking anymore.

BOB WALLACE: No, no, no. Shareware is a marketing hack. *(laughter, applause)*

WOZNIAK: Somebody who's designing something for himself has at least got a market of one that he's very close to.

ANDREW FLUEGELMAN *(author and distributor of PC-TALK[2], an excellent telecommunications program for IBM PCs and compatibles; founding editor of* PC World *and* Macworld*)*: That's what got me started. I originally wrote PC-TALK as a pure hack. I won't confess what language I wrote it in, but the fact is that I had owned my computer for about a month and I was trying to send my files to someone using a completely different computer, and there was not one piece of software in the entire world that would let me do that. I stayed up for a lot of nights to figure out a way to do it, and I consider that to be very much within the hacker ethic or spirit.

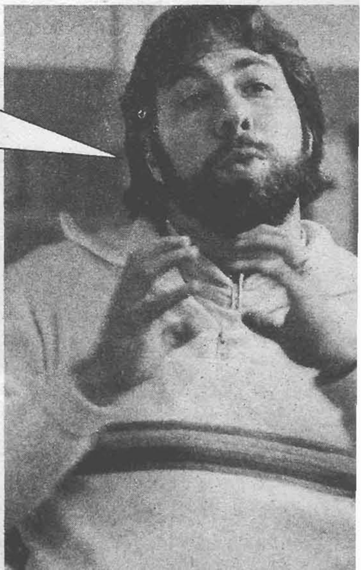
What got me away from being a hacker was when I figured out, "How can I get this out to people?" Although I'm known for giving away software for free, I did it purely to figure out how I could make some money with what I had done. The reason it's been successful is very strange. On the one hand, what people buy is not really access to the program, or the information. What they're mainly buying is the support, the stability, and the fact that it works reliably. And the reason for that is because I've had the opportunity to get a lot of feedback from a lot of people who were pissed off when they got Version 1.6 of the program, found that it didn't work with their modem, and they called me and said, "Hey, I've got this strange situation and here's what you can do to fix it."

I call that "freeback," and that's really what made the program successful. Right now my highest cost is user support. More than half of all the money I spend is to have people on the phone telling, not programmer types, but just regular people, how to use the program. In that respect my business looks similar to very commercial ventures. The difference is that it's been made accessible to people in a very unconditional way, and that's what people have responded to.

DAVID LUBAR *(game designer for Activision)*: You don't have to say that you *either* give it away or sell it. For example, a while back, just for the fun of it, I tried to see if I could compress Apple pictures and I came up with some code that required less disk space, so I published the listing in a magazine and as a result other people looked at it and said, "Hey, here's a better way," and it evolved through a whole bunch of people coming up with more and more compression. At the same time I gave the code itself to a publisher who put it out as part of a package and I get nice royalties from it. So it's not one world or the other.

DOUG CARLSTON: I think that there's a certain level of naivete here about the commercial world as a whole. All you have to do is take a look at the Japa-

**"In a company, sometimes a product gets developed and the company decides it doesn't fit a market, it won't sell. But because they own the product, they will squash it and say, 'Even though we're not gonna put it out, nobody else in the world's gonna get it.' That's a hiding of information, and that is wrong.** —*STEVE WOZNIAK, designer of Apple computer, co-founder of Apple Computer, Inc.*



[2] PC-TALK.III, $35 suggested donation, from Freeware/ Headlands Press, Box 862, Tiburon, CA 94920. Available free through most users' groups.

nese Ministry of Industry and Trade, MITI. Japan certainly has gotta be one of the most commercial nations on Earth. With software they essentially wanted to require anybody who owned any proprietary product to license it to anybody who felt that they had a need for it, and if they refused such a license, it would then be stripped of its copyright protection. That's because Japan feels that it has a strong competitive advantage in the manufacture and sale of hardware, but they feel like they're years behind in the development of software, and frankly what they really wanted to do was strip the advantage that other nations had in the development of software from them so that they could take it if they wanted to use it.

The dissemination of information as a free object is a worthy goal, it's the way most of us learned in the first place. But the truth of the matter is, what people are doing has more and more commercial value and if there's any way for people to make money off of it, somebody's gonna try to get an angle on it. So I think that it ought to be up to the people who design the product whether or not they want to give it away or sell it. It's their product and it should be a personal decision.

BILL ATKINSON *(author of MACPAINT, the landmark graphics program on the Macintosh):* Sometimes it's not even money. When I was working on QUICKDRAW I came across some improvements, real good algorithms, that I'd never seen before that I would love to tell lots of people about, because I think they're really neat hacks. And yet, I want to see Apple around in twenty years. It's not money for me; they're not paying me money to not talk about QUICKDRAW. I just know there's something there that gives Mac an advantage over an IBM PC and I don't really want to see IBM rip off QUICKDRAW. I don't. *(applause)*

ANDREW FLUEGELMAN: The problem is just distribution. There's been no thing on Earth so easy to distribute to people as software. The reason that we go out and pay five or ten or twenty thousand dollars for a car is because you need a key to drive it and it's usually sitting behind a fence at the dealer's showroom. The fact is that no matter how hard we work on something, how much inspiration or insight we put into it, once it's completed, in the medium in which we work, it's a trivial matter to make a perfect copy and give it to anybody in the world, instantly. That's what's been challenging us. It has nothing to do with whether it's easier or harder to make cars or write programs.

TED KAEHLER *(programmer at Xerox PARC — Palo Alto Research Center):* Do you think it's reasonable, through the scheme you're using to support Freeware, that everyone in this room could be making a living that way?

FLUEGELMAN: I really don't know. I did it just as a giggle. The reason I started was because I'd finished this program, I was gonna send it out, and I knew that I didn't have a prayer of coming up with a copy-protection scheme that some kid in San Diego wasn't gonna break the first night, so I figured I've gotta work with the system somehow.

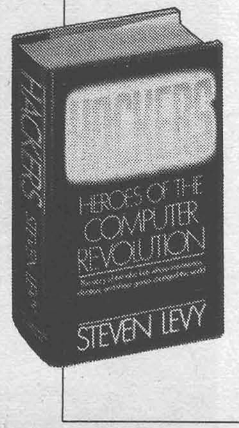
KAEHLER: You must know something about whether or not this many people could be doing that.

FLUEGELMAN: I think maybe, if a lot of people were willing to put out what I would call fully supported programs. That means not just something that gets the job done for you, which is what I did in the first round, but one that is error-trapped, that is documented, that is supported, that looks like it's been given all those trappings of value. Then maybe you can appeal to people's sense of value and they'll contribute for it. It's worked for me, I know it's worked for Bob and for a few others.

DOUG GARR *(journalist from* Omni*):* Could you tell us how the economics of Freeware works?

FLUEGELMAN: I send out the program and I ask for a $35 contribution, which for a program of its type many people say is one-fifth the cost of what they'd expect to spend commercially. So it's a bargain to begin with. I encourage people to make copies. I try and discourage people from re-selling the program and large corporations from making thousands of copies. I tell people that whether they liked it or not, give it to a friend and if their friend likes it then maybe they'll send me some money. I would guess that about one-tenth of the people who are using the program now have paid for it, and there are a lot of commercial software companies that can't make that claim. *(laughter)*

## Hackers

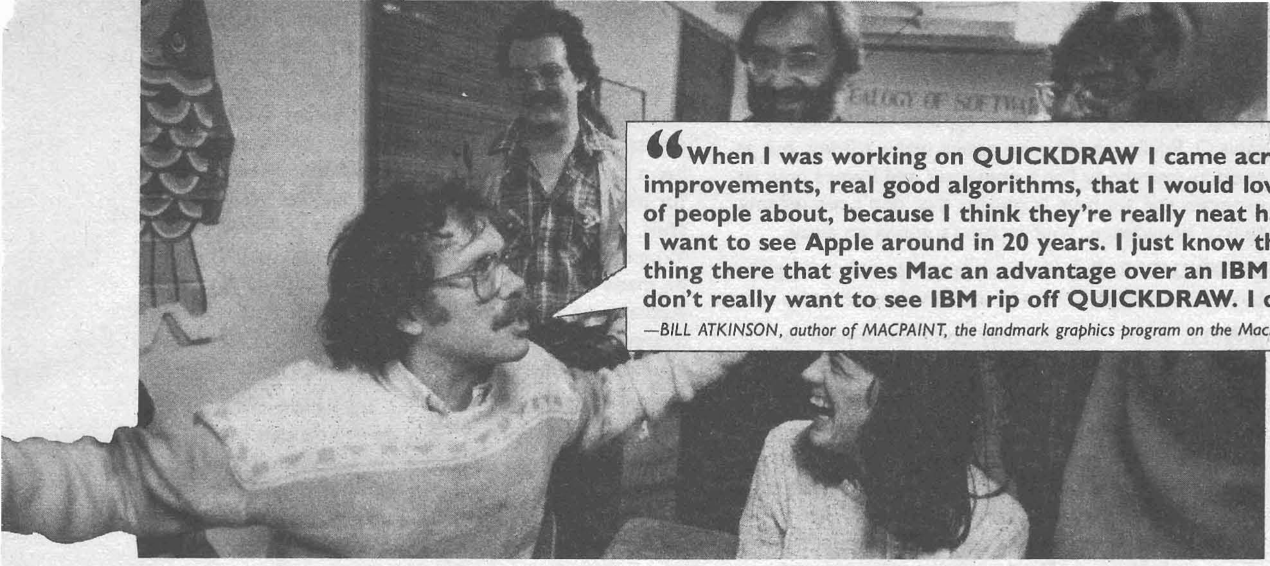


*The founding text for the Hackers' Conference was Steven Levy's* **Hackers** *(1984; 458 pp.; $17.95 from Doubleday and Company, 501 Franklin Ave., Garden City, NY 11530, or Computer Literacy).*

*Levy does for computers what Tom Wolfe did for space with* The Right Stuff. *Both are behind-the-scenes tales of elite athletes pursuing potent new technologies; both are vividly written; both are inspiring.*

*The very structure of the book was the occasion for the Conference. Levy chronicles three generations of hackers — the mini-computer all-night coders at MIT and Stanford in the '60s, the hardware hackers around the Homebrew Computer Club who made the first personal computers in the mid-'70s, and the myriad home-grown programmers on those computers as soon as they hit the market, who gave us the galaxy of consumer software from VISICALC to CHOPLIFTER. In the succession of generations Levy portrays a gradual degrading, commercializing of the Hacker Ethic.*

*The Hackers' Conference was called to join the three generations for the first time to see if they had anything to say to each other, and to see where the Hacker Ethic really was after years of stress in the boom-and-bust computer business. "Each generation," remarked conference co-designer Lee Felsenstein, "has suffered an infusion of Big Money. It may be interesting for them to compare how they've dealt with that."* —SB



> **“When I was working on QUICKDRAW I came across some improvements, real good algorithms, that I would love to tell lots of people about, because I think they're really neat hacks. And yet, I want to see Apple around in 20 years. I just know there's something there that gives Mac an advantage over an IBM PC and I don't really want to see IBM rip off QUICKDRAW. I don't.”**
>
> *—BILL ATKINSON, author of MACPAINT, the landmark graphics program on the Macintosh*

STEVEN LEVY: There's someone here who's supporting a program that doesn't ask for money. Dennis Brothers, do you want to tell us about MACTEP and what you've done there?

DENNIS BROTHERS: It's kind of a strange situation. I wrote it for my own use. I needed a communications program for the Macintosh, so I wrote it, and it turned out to be something that a lot of other people wanted as well. It's very primitive, very crude, compared to PC-TALK, but it was the right place and the right time, and there was tremendous response for it. I'm kicking myself a little now; maybe I should have put a little message in there: “Please send 35 bucks.” *(laughter)*

FLUEGELMAN: I just want to know: How many people in this room are using Dennis's program and would send him some money for it? I would.

VOICE: Why don't you ask it as two questions? *(laughter)*

FLUEGELMAN: No, it's a compound question.

BROTHERS: It is not a high enough quality program, in my estimation, to warrant that. And I don't have the time to put into it to bring it up to the level of PC-TALK where I believe it would be worth that kind of contribution.

ART KLEINER *(telecommunications editor for* Whole Earth Software Catalog *and* Whole Earth Review*)*: You had time to hang out on Compuserve [network] and answer people's questions, though.

BROTHERS: Yeah, but that's more for the fun of it. I don't have any better luck explaining this to my wife than explaining it to you guys. *(laughter)* Someday I may make most of my income off that program and its derivatives and related things, but today my primary business is completely unrelated to that, and I just don't have the time. I give what support I can, for much the same reason that I'm at this conference, for the interaction with other hackers over a network. I don't know, I'm having a little trouble in my own mind figuring out just why I did it the way I did it.

RICHARD STALLMAN: What would you think if someone else wanted to work on improving it, say, and then distributed it as freeware and split the results with you?

BROTHERS: It has happened and they are not splitting *(laughter)* and I don't know how to handle that.

BRIAN HARVEY: I'd like to argue against the idea of intellectual property in software. And here's why. I have a version of LOGO for UNIX that I worked on, that I wrote. So it's my intellectual property, right? I started with something that somebody else did and improved it. I improved it a lot; it's about 90 percent me. But I started with somebody else's structure. Now, before that *he* started with some terrific intellectual work done by Seymour Papert and Wally Fertzog and the gang at BBN [Bolt, Beranek & Newman, a Cambridge research institute] and MIT. I also started from the work done by Ken Thompson and Dennis Ritchey and Brian Kernighan to give me the programming tools that I needed to write that thing. I also started with a whole basis of material support from the guys who built the hardware and designed the hardware. Okay? That's not to say that I didn't do anything.

VOICE: Don't forget your mother and father. *(laughter)*

HARVEY: Damn straight. And the people who were paying my salary while I was doing it — they weren't paying me exactly to do that *(laughter)*, but hang on, the truth is I was a teacher in a high school and I needed this program to teach my kids. They weren't paying me to be a programmer, but I did it because it was something I needed to support my work. The point is what I did was based on the work of a hell of a lot of other people, all right? I think that's true of anything that anybody does. If I say fuck the world this is my thing and I'm in it for what I can get, then I'm a son of a bitch.

STEVE WOZNIAK: Philosophically you go higher and higher and higher and the whole world is the best thing. If the world gains, that's better than if your little country gains, or your little company gains. But then we don't want the others to get it, because “If IBM gets it it's gonna be a bad outcome for The People.” It turns out that that's either bullshit or something else, but it's bullshit. It turns out if IBM got it the rest of the world would really have more and do more. We really just want to make as much money as we can off of what we put our time in. Now you take that one level further and . . . I forget what I was gonna say. *(laughter, applause)*

JERRY JEWELL *(founder of Sirius Software, publisher of computer games)*: I think in most cases the programmers here who are wanting to make money at this are a lot like old witchdoctors. As long as they can keep a secret how they do things, it appears to be magic to John Q. Public, and they're gonna make a living, but as soon as everybody has a computer and knows how to program and we have languages that don't require any special knowledge, your income's gonna go away.

DAVID LUBAR: But there are more people willing to buy games and play them than are willing to write them.

JEWELL: Right. Because they don't know how to write them.

WOZNIAK: I remember what I was gonna say. The company wants to keep it secret to make as much money as they can, but here's how we get beyond that level. We say that the whole world wins because other people are more inspired to go write their own programs and design their own hardware because they're gonna make money. They're gonna make so much product and do so well off it that they'll go out and do the most incredible things. They're inspired. That's the American way.

RICHARD GREENBLATT: There is a force in this world for standardization. If there's a knowledgeable marketplace people will say, "Gee, we want to do things a standard way." That's what IBM really did right. They said, "We're gonna have an open architecture on the PCs," and they advertised that and it was the one thing they did right, and look where it got 'em. In software that same thing can happen. If you have something done right and it's standardized and it's public, people will want that as opposed to the proprietary thing. And it's not necessarily because it's better today than the proprietary thing, but they realize that it is building a foundation and over the long term maybe it will get to be better than the proprietary thing.

WOZNIAK: Customers set the standards.

GREENBLATT: Customers inevitably will set the standards, no matter what.

DAVE HUGHES *("Sourcevoid Dave," system operator of pace-setting bulletin board system — 303/632-3391):* Hackers are doomed, and you just better accept that. *(Hssssss)* Not doomed to extinction, you're doomed to live a life in which you're on the frontier. Nobody pays for my WORD-DANCE, nobody paid for your early stuff, nobody paid for T. S. Eliot's first goddamn poems. When he got commercial, then the ethic meant when he made it he damn well better cycle back, and at least Apple and a few companies try to give it back, and the Shareware and Freeware is an attempt to try to reconcile that boundary toward an ethic and a commitment.

HENRY LIEBERMAN *(MIT A.I. Lab):* How does the frontier get supported? How do the centers of research and the centers of education get supported? I think there is another kind of software piracy going on that's not discussed very much, and the villains are not high school kids who copy discs and break secret codes. They're executives in three-piece suits that work for large corporations and drive Mercedes. They make money off the results of research and education, and they don't kick very much back to support the next generation.

VOICE: They will argue that they paid the taxes that funded the MIT A.I. Lab.

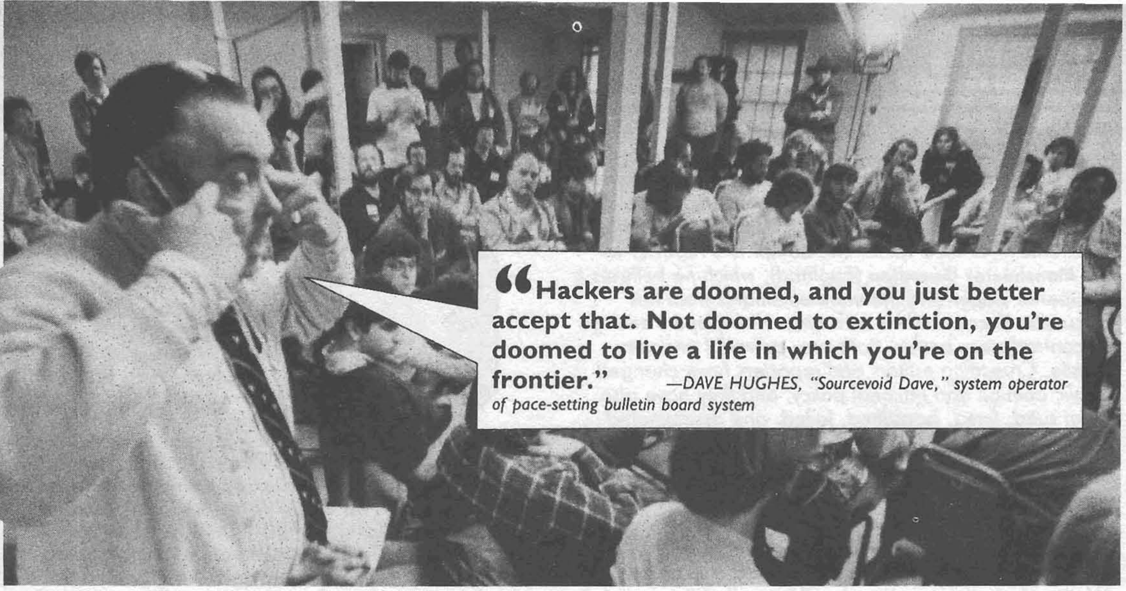
LIEBERMAN: That's true, and that is the only reason that places like MIT and Stanford don't disappear entirely off the face of the Earth. We have this paradoxical situation where the computer industry is booming and yet places like MIT and Stanford don't have secure support. It's very likely that I will be out of a job in a year. Places like the MIT A.I. Lab get no direct benefit from places like IBM or Apple. Well, that's not true, that's not true. They give us discounts on their machines, and that's very helpful.

And they contribute some cash, but the amount they contribute is piddling in the sense that when it comes time to pay my salary, the people I work for have to go begging to people like ARPA and they have to promise to build bombs *(murmuring)* [ARPA is Advanced Research Projects Agency, part of the Defense Department] and that disturbs me deeply. I and my colleagues come up with important ideas which people acknowledge helps support the industry and makes money for people. I would like to be able to pursue my

*Left, John Draper (Cap'n Crunch) with one of the original "blue boxes" used for hacking up the planet's telephone system.*

*Above, Scott Kim, author of* Inversions, *designer of the Hackers' Conference logo and T-shirt (p. 44).*

*Right: Diana Merry, Xerox PARC.*



> “Hackers are doomed, and you just better accept that. Not doomed to extinction, you’re doomed to live a life in which you’re on the frontier.” —*DAVE HUGHES, “Sourcevoid Dave,” system operator of pace-setting bulletin board system*

work without having to go to the Defense Department.

RICHARD STALLMAN: It’s worse than that even, because at a university paid for by everyone in the country an idea will be developed almost to the point where you can use it, but then the last bit of work will be done by some company and the company will get lots of money, and those of us who already paid for most of the work won’t be able to use the results without paying again, and we won’t be able to get the sources even though we paid for those sources to be written.

LES ERNEST *(founder of Imagen Systems, former head of Stanford A.I. Lab)*: Various ideas have been given about what is the essence of hacking. Is it altruism or is it financial motive? My view is that it’s primarily an ego trip, by most people. All good hacks are done by somebody who thinks he can do it a lot better than anybody else, and he goes off and does it. There are very few team hacks that one can think of that went anywhere. *(murmuring)* Of course commercial development is intrinsically a team effort, and therefore there is always some tugging going on when you change over from being a hacker to trying to do some commercial development. It was mentioned a little while ago that Japan, while they have good hardware, don’t seem to have good software for the most part. My view is, that’s a cultural problem; Japanese culture values team effort very much; it does not value ego trips.

BILL BURNS: I think Les is right, and I also agree with what Woz said, and I would like to propose that we separate two things. I think the “hacker drive” is individual, it’s a drive within us. It’s what happens when we’re doing something absolutely useless; we just decided to tickle a line of code and see where it went at some weird 3 A.M. on a Saturday morning. But then what happens to the product of that is a whole ’nother set of questions. I think if we can separate the hacker drive from the products of hacking, which can either have no economic value or tremendous economic value but still have the same hacker value, then I think the discussion will get a little farther.

LEE FELSENSTEIN *(designer of Osborne I, co-founder of Community Memory)*: If you’re only dealing with one of those two things in your life, if you define yourself in only one area, you are crippled, I say. I’ve seen a lot of cripples on the other side, too. If you’re only taking stuff that other people make, and playing games with it to somehow get money out of it — I believe that people like that (of course, I’m not one of them) *(laughter)* . . . people like that know that they’re the ones that are playing the win/lose game. “If I give it to you I must take it from him.” And that results in what I and other people call the “production of scarcity.”

We have a responsibility to know about and live to a certain extent on the other side of the fence and find out what happens with these things once they’re produced. And we should also expect the people who live most of their lives over there to come onto our side and learn to play a little bit, learn to express some of their own creativity. Concentrating on one thing alone makes you into a deformed person.

DICK HEISER *(owner and proprietor of the original Computer Store in 1975, now with Xanadu)*: It seems like you can have a variable amount of your own content in something. As a computer retailer I found myself turning over other people’s goods. I wanted to distinguish the quality of my service, but I found that hard. Don Lancaster, who wrote a book called *The Incredible Secret Money Machine (applause)*, talked about the fact that if you are maximizing the added value, rather than trying to leverage other people’s money or other people’s work, then a miracle can go on.

You have to be committed, and you don’t know how it’s gonna work out, but the amount of power factor that’s going on in this technology is so astounding that you ought to be encouraged to try. In this miraculous environment, we find people like Bob Wallace doing things that succeed very much beyond the expectations that he probably had. Similarly my store started out as kind of a hacker-type thing that became much too commercial. You keep deciding, “Is that what I want?”

You keep designing, you keep adding personal value, and the miracle keeps happening. It doesn’t happen for everybody, unfortunately, and it doesn’t happen automatically, but if you’re willing to experiment, and if you beware of too much money and too many other people getting involved, so that you can make your own decisions, then you’re free to try these wonderful things and see if they work. And sometimes they do. ■