Editorial

How Can We Stop Software Piracy?

Chris Morgan, Editor in Chief

Software piracy is rapidly becoming a major problem in the personal computer field. The casual copying of programs by computer hobbyists, although not at the epidemic stage, is frighteningly commonplace. Many people fail to see (or prefer not to see) that the practice is not just illegal—it's *unethical.*

But what about making backup copies of important software? What happens if your small business' direct-mail program "dies"? Without a backup, a businessman's only recourse is to return the disk to the manufacturer and hope it won't take longer than a few weeks to get a replacement. Manufacturers understand the problem, and have designed some floppy-disk-based programs that allow the user to make one backup copy. After this, software "jamming" information is automatically added to the original floppy disk to theoretically prevent additional illegal copies. In practice, though, enterprising software experts can crack the protection mechanisms and make copies at will.

The industry is faced with a dilemma: how does the manufacturer serve the customer's legitimate need to make backup copies, while protecting his expensive software investment? There are two possibilities: put the would-be software pirate at a disadvantage if he makes an illegal copy, or, better still, make it virtually impossible for the pirate to make a copy.

The Persuasion Route

Let me make a not-too-perfect analogy between the software industry and the record industry. When tape recorder sales began to increase during the early 1970s, record industry executives predicted that record sales would plummet because of private off-the-air taping. But, in fact, record sales climbed steadily throughout the decade. Why? My opinion is that when people think of a recording, they think of the entire package: the album artwork, the liner notes—in short, there is more to a recording than the sound coming from a pair of loudspeakers. In much the same vein, there is more to a piece of software than the object code: there is the documentation, for instance.

The need to make a copy of the documentation is an additional nuisance for the software pirate. It costs money to make photocopies. Then there's the registration card: legitimate owners of software are often put on mailing lists to receive updates to their programs as well as information about new programs from the manufacturer. A cheap and effective way for manufacturers to fight the pirate is to creatively exploit the latter idea. At the risk of overgeneralization, computer-science people tend to be obsessive-compulsive in their psychological makeup, ie: they hate to miss out on any details about a product they buy—especially a piece of software!

I mentioned earlier that this was a less-than-perfect analogy. The problem is that a $9.95 recording is one thing—a $600 program is quite another. The above-mentioned tactics might help the manufacturer of a $30 or $50 piece of software, but temptation becomes powerful indeed when the price tag reaches three or four figures.

Editorial continued on page 10

Technological Measures

The ultimate answer is to make it so difficult and costly for the pirate to make copies that the problem goes away. A good first step is to put teeth into software protection laws. The revised copyright act of 1976 had a major impact on phonograph record pirates because of the much more stringent penalties for convicted offenders. You may have noticed the Ⓟ sign on commercial records and tapes: it's an indication that they're protected by the new law. (For further legal background, including information on the latest Supreme Court decisions, see "Washington Tackles the Software Problem," page 128, and "Legal Protection for Computer Hardware and Software," page 140.)

We come next to the most intriguing weapon in our arsenal: hardware "locks" on the software. The concept of the *I.D. ROM* is a recent development now being used, among other places, in conjunction with a program called RCS/Micro Modeller, developed in England by Intelligence (UK) Limited. The program allows a person to use an Apple II computer to create financial planning models and high-resolution color displays featuring pie charts, histograms, and so on. A novel feature of the program is its "electronic slide show" capability: a hand-held control, similar to a slide projector control, plugs into one of the paddle ports of the Apple and allows the user to cycle through an electronic "slide show" on the video screen. Built into the control is a special ROM containing an identification number that is duplicated on the program floppy disk. The program periodically checks for the presence of the I.D. ROM. If it's not found, the program crashes.

This technique puts one more stumbling block in the way of the pirate, and it does not add appreciably to the total cost of the software (the I.D. ROM costs about $20). Alas, there are some experts in Europe who have cracked the code of another I.D. ROM used in conjunction with a program called Wordcraft, which is being distributed by Commodore in England. So the technique, while making it much more difficult to copy software, is not the ultimate answer. Still, I welcome this type of innovative approach to a mind-boggling problem. Readers interested in further information about the RCS/Micro Modeller program (not yet available in the United States) should contact David Low, ACT (Microsoft) Ltd, 5/6 Vicarage Rd, Edgbaston, Birmingham B15 3ES England.

Two of the most promising solutions to the software protection problem come from West Coast inventor Marc Kaufman. He has filed a patent for an "execute-only ROM," a new type of read-only memory which produces a sequence of executable code in the normal manner, but prohibits the user from randomly accessing memory addresses. As Kaufman explains, the user begins execution of the program at a known address. A "secret" executive routine, built into the ROM, contains a table of the legal next steps for every given step in the program. Only those steps listed in the table can be accessed by the user. For example, if the program contains a branch to one of two places, *only* those two places can be examined by the programmer at that time. If a program contains enough branches, it would take an inordinate amount of time for the user to run through every permutation of the program to get a complete listing of the code, even if a computer did the searching. Kaufman is presently working with both hardware vendors and users to develop the idea. An unreadable EPROM is also in the works, enabling the do-it-yourselfer to create secure programs.

Kaufman's second idea is to add a "black box" to a personal computer. Every piece of software would come with a magnetic key (or other type of hard-to-duplicate key) that plugs into the black box and contains a coded I.D. number that matches the I.D. number on the floppy disk. The program resides on the disk in encrypted form. In order to decode the program, the key must be plugged into the box. With this scheme, the user can make as many backup copies as desired, but only one of them can be used at a time. The drawback to such a system is the need for the black box. But if the idea catches on, the price would probably come down. Interested readers can contact Marc Kaufman at Kaufman Research, 14100 Donelson Pl, Los Altos Hills CA 94022.

Stopping the pirate is vital. Piracy has reached near epidemic levels in Europe, where it is not uncommon for an entire computer club numbering in the hundreds to line up their computers and make hundreds of copies of programs from United States manufacturers for the use of the entire club! Then there is the phenomenon of the "software library." Some of them are legitimate, but all too many cavalierly offer copies of programs to their members at a fraction of the retail cost.

Illegitimate copies of programs threaten the fabric of personal computing. The software innovators in our field must be compensated fairly for their work, or we will no longer see the high-quality programs that currently grace the marketplace.

I welcome comments from readers about this all-important issue, and would like to begin a dialog featuring your comments. Please send your thoughts to: Software Protection, c/o BYTE Publications Inc, POB 372, Hancock NH 03449. ■

Articles Policy

BYTE is continually seeking quality manuscripts written by individuals who are applying personal computer systems, designing such systems, or who have knowledge which will prove useful to our readers. For a more formal description of procedures and requirements, potential authors should send a large (9 by 12 inch, 30.5 by 22.8 cm), self-addressed envelope, with 28 cents US postage affixed, to BYTE Author's Guide, 70 Main St, Peterborough NH 03458.

Articles which are accepted are purchased with a rate of up to $50 per magazine page, based on technical quality and suitability for BYTE's readership. Each month, the authors of the two leading articles in the reader poll (BYTE's Ongoing Monitor Box or "BOMB") are presented with bonus checks of $100 and $50. Unsolicited materials should be accompanied by full name and address, as well as return postage.