The HACKER *Ethic*

and the Spirit of the Information Age

PEKKA HIMANEN

CHAPTER 4

The Academy and the Monastery

The Open Model

In the original hacker money ethic, the new economy's governing attitude, "which seeks profit rationally and systematically" (Weber's description of the spirit of old capitalism, which still applies well to our time),[1] is challenged by the open model in which the hacker gives his or her creation freely for others to use, test, and develop further. For the original MIT hackers, this idea was even as defining an element of the hacker ethic as the hacker relation to work, but nowadays the "jargon file" says that this ethical ideal of openness is accepted among hackers "widely, but not universally."[2]

Although from this book's perspective the ethical arguments of hackerism are the most interesting and important ones, there is also a more pragmatic level that is significant and fascinating. Just as we can add to our ethical ar-

guments for the passionate and free work ethic the more pragmatic point that, in the information age, new information is created most effectively by allowing for playfulness and for the possibility of working according to one's individual rhythm, we can likewise say that the open model is not just ethically justified but also very powerful in practice. (In fact, the "jargon file" also says that it is a "*powerful* positive good.") It is worth taking a closer look at the hackers' idea of openness from this viewpoint. The development of the Net would be a great example, but the Linux project, which has arguably taken the ideal of openness the furthest so far, serves as an even better one. After understanding this powerful model that has made the Net and Linux possible, we can think of some ways in which the open model could be applied to areas of life other than software.

Torvalds started working on Linux in 1991 while he was a student at the University of Helsinki.[3] After developing an interest in the problems of operating systems, Torvalds imported into his home computer the Unix-like Minix operating system, written by Dutch computer-science professor Andrew Tanenbaum and, by studying and using it as a developmental framework, proceeded to design his own one.[4] An essential feature of Torvalds's work was that he involved others in his project from the very beginning. On August 25, 1991, he posted a message on the Net with the subject line "What would you like to see most in minix?" in which he announced that he was "doing a

(free) operating system."[5] He received several ideas in reply and even some promises for help in testing the program. The operating system's first version was released on the Net as source code free to all in September 1991.[6]

The next, improved version was available as soon as early October. Torvalds then extended an even more direct invitation to others to join him in the development of the new system.[7] In a message sent to the Net, he asked for tips about information sources. He got them, and development advanced quickly. Within a month, other programmers had joined in. Since then, the Linux network has grown at an amazing creative pace. Thousands of programmers have participated in Linux's development, and their numbers are growing steadily. There are millions of users, and their number, too, is growing. Anyone can participate in its development, and anyone is welcome to use it freely.[8]

For the coordination of their development work, Linux hackers use the entire toolbox of the Net: e-mail, mailing lists, newsgroups, file servers, and webpages.[9] Development work has also been divided into independent modules out of which hacker groups create competing versions. A group consisting of Torvalds and a few other principal developers then decides which of these versions will be incorporated in the improved version of Linux (and, of course, the modular structure also develops gradually). Torvalds's group does not, however, hold any permanent position of authority. The group retains its au-

thority only for as long as its choices correspond with the considered choices of the hacker community. Should the group's choice prove less than enlightened, the hacker community proceeds to develop the project in its own direction, bypassing the former leaders of the pack.

In order to control the continuous development of Linux, publications have been divided into two series. In the stable versions, safe for use by average users, the *y* in the release number *x.y.z* is even (e.g., version 1.0.0), whereas in the developmental versions, aimed at programmers, the *y* is the stable version's *y* + 1 (e.g., the stable version 1.0.0's improved but still not finally tested developmental version is 1.1.0). *X* grows only when a truly fundamental change is made (at the time of writing, the latest available version is 2.4.0). This simple model has worked surprisingly well in the management of Linux development.

In the well-known essay "The Cathedral and the Bazaar," published originally on the Net, Raymond has defined the difference between Linux's open model and the closed model preferred by most companies by comparing them to the bazaar and the cathedral. Although a technologist himself, Raymond emphasizes that Linux's real innovation was not technical but social: it was the new, completely open social manner in which it was developed. In his vocabulary, it was the shift from the cathedral to the bazaar.[10]

Raymond defines the cathedral as a model in which one

person or a very small group of people plans everything in advance and then realizes the plan under its own power. Development occurs behind closed doors, and everybody else will see only the "finished" results. In the bazaar model, on the other hand, ideation is open to everyone, and ideas are handed out to be tested by others from the very beginning. The multiplicity of viewpoints is important: when ideas are disseminated widely in an early stage, they can still benefit from external additions and criticisms by others, whereas when a cathedral is presented in its finished form, its foundations can no longer be changed. In the bazaar, people try out different approaches, and, when someone has a brilliant idea, the others adopt it and build upon it.

Generally speaking, this open-source model can be described as follows: it all begins with a problem or goal someone finds personally significant. That person may release just the problem or goal itself, but usually he or she will also provide a Solution—version 0.1.1, to use the Linux numbering system. In the open model, a recipient has the right to freely use, test, and develop this Solution. This is possible only if the information that has led to the Solution (the source) has been passed on with it. In the open-source model, the release of these rights entails two obligations: these same rights have to be passed on when the original Solution or its refined version (0.1.2) is shared, and the contributors must always be credited whenever either version is shared. All this is a shared

process, in which the participants move gradually—or sometimes even by leaps and bounds (say, a shift from version 0.*y*.*z* to version 1.*y*.*z*)—to better versions. In practice, of course, projects follow this idealized model to a greater or lesser extent.

The Academy and the Monastery

Another possible allegory for the open-source model is again the academy, which it resembles even more directly than the cathedral. Scientists, too, release their work openly to others for their use, testing, and further development. Their research is based on the idea of an open and self-correcting process. The latter idea of self-correction was emphasized by Robert Merton as an equally important cornerstone of scientific ethic as openness. He called it *organized skepticism*[11]—historically, it is a continuation of the *synusia* of Plato's Academy, which also included the idea of approaching the truth through critical dialogue.[12] The scientific ethic entails a model in which theories are developed collectively and their flaws are perceived and gradually removed by means of criticism provided by the entire scientific community.[13]

Of course, scientists, too, have chosen this model not only for ethical reasons but also because it has proved to be the most successful way of creating scientific knowledge. All of our understanding of nature is based on this academic or scientific model. The reason why the original hackers' open-source model works so effectively seems to

be—in addition to the facts that they are realizing their passions and are motivated by peer recognition, as scientists are also—that to a great degree it conforms to the ideal open academic model, which is historically the best adapted for information creation.

Broadly speaking, one can say that in the academic model the point of departure also tends to be a problem or goal researchers find personally interesting; they then provide their own Solution (even though in many instances the mere statement of the problem or proclamation of a program is interesting in itself). The academic ethic demands that anyone may use, criticize, and develop this Solution. More important than any final result is the underlying information or chain of arguments that has produced the Solution. (It is not enough to merely publish "$E = mc^2$"—theoretical and empirical justifications are also required.) Nevertheless, the scientific ethic does not involve only rights; it also has the same two fundamental obligations: the sources must always be mentioned (plagiarism is ethically abhorrent), and the new Solution must not be kept secret but must be published again for the benefit of the scientific community. The fulfillment of these two obligations is not required by law but by the scientific community's internal, powerful moral sanctions.

Following this model, normal physics research, for example, continuously provides new additions ("developmental versions") to what has already been achieved, and after testing these refinements the scientific community accepts them as part of its body of knowledge ("stable ver-

sions"). Much more rarely, there is an entire "paradigm shift," to use the expression that philosopher of science Thomas Kuhn introduced in his book *The Structure of Scientific Revolutions*.[14] In the broadest sense, there have been only three long-lived research paradigms in physics: the Aristotelian-Ptolemaic physics, the "classic" Newtonian physics, and the Einsteinian-Heisenbergian physics based on the theory of relativity and quantum mechanics. Seen this way, present theories are versions 3.*y.z*. (Many physicists already call the version 4, which they believe is imminent, "The Theory of Everything." Computer hackers would not anticipate the arrival of version 4.0.0 quite so eagerly.)

The opposite of this hacker and academic open model can be called the closed model, which does not just close off information but is also authoritarian. In a business enterprise built on the monastery model, authority sets the goal and chooses a closed group of people to implement it. After the group has completed its own testing, others have to accept the result as it is. Other uses of it are "unauthorized uses." We can again use our allegory of the monastery as an apt metaphor for this style, which is well summed up by Saint Basil the Great's monastic rule from the fourth century: "No one is to concern himself with the superior's method of administration."[15] The closed model does not allow for initiative or criticism that would enable an activity to become more creative and self-corrective.

We have mentioned that hackers oppose hierarchical operation for such ethical reasons as that it easily leads to

a culture in which people are humiliated, but they also think that the nonhierarchical manner is the most effective one. From the point of view of a traditionally structured business, this may initially seem quite senseless. How could it ever work? Should not someone draw an organization chart for the Net and Linux developers? It is interesting to note that similar things might be said of science. How could Einstein ever arrive at his $E = mc^2$ in the chaos of self-organized groups of researchers? Should science not operate with a clear-cut hierarchy, headed up by a CEO of Science, with a division chief for every discipline?

Both scientists and hackers have learned from experience that the lack of strong structures is one of the reasons why this model is so powerful. Hackers and scientists can just start to realize their passions and then network with other individuals who share them. This spirit clearly differs from that found not only in business but also in government. In governmental agencies, the idea of authority permeates an action even more strongly than it does in companies. For the hackers, the typical governmental way of having endless meetings, forming countless committees, drafting tedious strategy papers, and so on before anything happens is at least as great a pain as doing market research to justify an idea before you can start to create. (It also irritates scientists and hackers no end when the university is turned into a governmental bureaucracy or monastery.)

But the relative lack of structures does not mean that

there are no structures. Despite its apparent tumult, hackerism does not exist in a state of anarchy any more than science does. Hacker and scientific projects have their relative guiding figures, such as Torvalds, whose task it is to help in determining direction and supporting the creativity of others. In addition, both the academic and hacker models have a special publication structure. Research is open to anyone, but in practice contributions included in reputable scientific publications are selected by a smaller group of referees. Still, this model is designed so as to guarantee that, in the long run, it is the truth that determines the referee group rather than the other way around. Like the academic referee group, the hacker network's referee group retains its position only as long as its choices correspond to the considered choices of the entire peer community. If the referee group is unable to do this, the community bypasses it and creates new channels. This means that at the bottom the authority status is open to anyone and is based only on achievement—no one can achieve permanent tenure. No one can assume a position in which his or her work could not be reviewed by peers, just as anyone else's creations can be.

The Hacker Learning Model

It goes without saying that the academy was very influential long before there were computer hackers. For example, from the nineteenth century onward, every industrial

technology (electricity, telephone, television, etc.) would have been unthinkable without its underpinning of scientific theory. The late industrial revolution already marked a transition to a society that relied upon scientific results; the hackers bring about a reminder that, in the information age, even more important than discrete scientific results is the *open academic model* that enables the creation of these results.

This is a central insight. In fact, it is so important that the second big reason for the pragmatic success of the hacker model seems to be the fact that hackers' learning is modeled the same way as their development of new software (which can actually be seen as the frontier of their collective learning). Thus, their learning model has the same strengths as the development model.

A typical hacker's learning process starts out with setting up an interesting problem, working toward a solution by using various sources, then submitting the solution to extensive testing. Learning more about a subject becomes the hacker's passion. Linus Torvalds initially taught himself programming on a computer he inherited from his grandfather. He set up problems for himself and found out what he needed to know to solve them. Many hackers have learned programming in a similarly informal way, following their passions. The example of the ability of ten-year-olds to learn very complicated programming issues tells us much about the importance of passion in the learning process, as opposed to the slow going their contem-

poraries often find their education in traditional schools to be.[16]

Later on, the beginnings of Torvalds's operating system arose out of his explorations into the processor of the PC he purchased in 1991. In typical hacker fashion, simple experiments with a program that just tested the features of the processor by writing out either *A*s or *B*s gradually expanded into a plan for a Net newsgroup-reading program and then on to the ambitious idea of an entire operating system.[17] But even though Torvalds is a self-taught programmer in the sense that he acquired his basic knowledge without taking a class, he did not learn everything all by himself. For example, in order to familiarize himself with operating systems, he studied the source codes of Tanenbaum's Minix as well as various other information sources provided by the hacker community. From the very beginning, in true hacker fashion, he has never hesitated to ask for help with questions in areas in which he has not yet acquired expertise.

A prime strength of the hacker learning model lies in the fact that a hacker's learning teaches others. When a hacker studies the source code of a program, he often develops it further, and others can learn from this work. When a hacker checks out information sources maintained on the Net, he often adds helpful information from his own experience. An ongoing, critical, evolutionary discussion forms around various problems. The reward for participating in this discussion is peer recognition.

The hackers' open learning model can be called their "Net Academy." It is a continuously evolving learning environment created by the learners themselves. The learning model adopted by hackers has many advantages. In the hacker world, the teachers or assemblers of information sources are often those who have just learned something. This is beneficial because often someone just engaged in the study of a subject is better able to teach it to others than the expert who no longer comes to it fresh and has, in a way, already lost his grasp of how novices think. For an expert, empathizing with someone who is just learning something involves levels of simplification that he or she often resists for intellectual reasons. Nor does the expert necessarily find the teaching of basics very satisfying, while a student may find doing such teaching tremendously rewarding, since he or she does not as a rule get to enjoy the position of instructor and is generally not given sufficient opportunity to use his or her talents. The process of teaching also involves by its very nature the comprehensive analysis of subject matter. If one is really able to teach something to others, one must have already made the material very clear to oneself. While preparing the material, one has to consider it carefully from the point of view of possible further questions and counterarguments.

Once again, this hacker model resembles Plato's Academy, where students were not regarded as targets for knowledge transmission but were referred to as compan-

ions in learning (*synetheis*).[18] In the Academy's view, the central task of teaching was to strengthen the learners' ability to pose problems, develop lines of thought, and present criticism. As a result, the teacher was metaphorically referred to as a midwife,[19] a matchmaker,[20] and a master of ceremonies at banquets.[21] It was not the teacher's task to inculcate the students with preestablished knowledge but to help them give birth to things from their own starting points.

In the hacker community, too, the experts understand themselves as learners who can just act as gadflies, midwives, and symposiarchs to others, thanks to their deeper knowledge.

The Net Academy

The ethos of the original academic and the hacker model—well summed up by Plato's idea that "no free person should learn anything like a slave"[22]—is totally different from that of the monastery (school), the spirit of which was summed up by Benedict's monastic rule: "It belongeth to the master to speak and to teach; it becometh the disciple to be silent and to listen."[23] The irony is that currently the academy tends to model its learning structure on the monastic sender-receiver model. The irony is usually only amplified when the academy starts to build a "virtual university": the result is a computerized monastery school.

The scientific revolution in the seventeenth century was supposed to mean the abandonment of scholasticism and its replacement with a science continually striving for new knowledge. Nevertheless, the university has preserved the scholastic teaching model and hierarchy, down to its vocabulary (e.g., a "dean" was originally an office-holder of a monastery). The scientific revolution took place four hundred years ago, but it is not very well reflected in our universities as a basis for research-based learning. It seems quite strange that we expect scholastic teaching methods to be able to produce modern individuals capable of independent thought and the creation of new knowledge.

The wider significance of the hacker learning model is its healthy reminder to us of the potential in the original idea of seeing the academic development and learning models as identical. We could also use this idea to create a generalized Net Academy, in which all study materials would be free for use, critique, and development by everyone. By improving existing material in new directions, the network would continuously produce better resources for the study of the subjects at hand. Members of the network would be driven by their passions for various subjects and by the peer recognition for their contributions.

Logically, the continued expansion and development of this material, as well as the discussion and examination of it, would also have to be the Net Academy's only way to grant study credits; and, true to the spirit, the highest

credits should be given for those accomplishments that prove the most valuable to the entire learning community. A hacker-style reading of the material with a view toward criticizing and improving it—that is, toward doing something, motivating oneself, with it—would also be much more conducive to learning than the current tendency to just read material.

The Net Academy would follow the hacker model in creating an important continuum from the beginning student all the way to the foremost researcher in the field. Students would learn by becoming researching learners from the very beginning, by discussing matters with researchers, and later on by studying the research publications of their field directly.

In the Net Academy, every learning event would permanently enrich all other learners. Alone or in the company of others, the learner would add something to the shared material. This differs from our present mode of disposable learning, in which every student starts from the beginning, passes the same exams isolated from everyone else, and never gets to benefit from the insights of others. Worse, after the exam the examiner basically tosses all those individual insights into the wastebasket. This is as absurd a procedure as would be the decision of each generation of researchers to finally toss all their results away ("I see, $E = mc^2$; so what—toss!") and let the next generation start over.[24]

It goes without saying that the practical realization of

the general Net Academy presents a great challenge. For example, as in the world of hackers and researchers, a guiding structure for the collective creation of learning materials is needed. When material is constantly adapted and expanded in new directions, competing versions are born. This is always the case in the hacker and research fields. Hackers have solved practical problems arising from this by developing so-called concurrent-versioning systems: these enable one to see how competing versions differ from the existing version and from each other. On a more theoretical level, the problem can be solved by the practice of referees. With the help of a concurrent-versioning system, a self-organized group of referees can make decisions between competing versions and combine their ideas if need be.

After the hackers' reminder of the full significance of the academic model, it would be odd to continue our current practice of providing learners mainly with results, without making them learn much more deeply the academic model itself, which is based on a collective process of posing of problems, the questioning of them, and the development of solutions—a process driven by passion and recognition for socially valuable contributions. The core of the academy does not consist of its individual achievements but of the academic model itself.

The Social Model

Expressing this one possible wider application inherent in the hacker model must not, of course, be understood to say that we should just wait for governments or corporations to execute it. A central point of hackerism is to remind us that through the open model great things can be accomplished by individuals' direct cooperation. The only limit is our imagination. For example, the hacker open model could be transformed into a social model—call it the open-resource model—in which someone announces: I have an idea, I can contribute this much to it, please join me! Although this version of the open model would also involve local physical action, the Net would be used as an effective means for joining forces and later disseminating and developing the idea further.

For example, I could announce on the Net that I would be willing, once in a while, to help some elderly person take care of things. I can announce that kids can come and play at our house after school. I can say that I would be glad to walk one of the neighborhood dogs on weekdays. Perhaps the effectiveness of this model could be strengthened by adding a condition that the helped person commit to helping someone else equally. The Net can be used as a means to organize local resources. Gradually, others will join the realization of great social ideas, and this will generate even greater ideas. There would be a self-feeding effect, as in the computer hacker model.

We have seen that the hacker model can bring about great things in cyberspace without governments and corporations as mediators. It remains to be seen what great things individuals' direct cooperation will accomplish in our "flesh reality."

PART THREE

The NETHIC

CHAPTER 5

From Netiquette to a Nethic

Netiquette and Nethic

Beyond the hacker work and money ethic is the significant third level of the hacker ethic that can be called the *nethic* or network ethic. This expression refers to the hackers' relationship to our network society's networks in a wider sense than the more familiar term *netiquette* (which concerns behavioral principles for communication on the Net—e.g., "avoid flaming," "read the file of frequently asked questions before posting your message," etc.).[1] Again, not all hackers share all the elements of the nethic, but still these elements are linked together in their social meaning and relation to the hacker ethic.

The first part of the hacker nethic consists of the hackers' relation to media networks such as the Net. Although we can say that hackers' characteristic relation to them dates back to the origin of the hacker ethic in the sixties,

this nethic has received a more conscious formulation in recent years. One key moment came in 1990 when hackers Mitch Kapor and John Perry Barlow started the Electronic Frontier Foundation in San Francisco to promote the fundamental rights of cyberspace.[2] Barlow, a child of the sixties counterculture, used to write songs for the Grateful Dead and became a pioneer in the cyber-rights movement. He was the first to apply William Gibson's term *cyberspace* (from his novel *Neuromancer*) to all electronic networks.[3] Kapor was an important player in the development of personal computers, creating, in 1982, the spreadsheet program Lotus. It was the first PC application that made a widespread function significantly easier than it had been before, and this made it an important factor in the breakthrough of the personal computer.[4] The name *Lotus* reflected Kapor's background: as a former mental-health counselor with a psychology degree, and later a transcendental-meditation instructor, he was interested in Eastern thought systems.

The enterprise, also called Lotus, that Kapor had built around his program quickly evolved into the largest software company of its time. But as his original hackerism became more and more entrepreneurial, Kapor began to feel alienated, and he left the business after four years. In his own words: "It felt awful to me, personally. So I left. I just walked away one day. . . . The things that were important to the business as an organism were things that I could demonstrate less and less enthusiasm for."[5]