# Preface to The Hacker Ethic

*This is an excerpt from Pekka Himanen's book* 'The Hacker Ethic'.

At the core of our technological time stands a fascinating group of people who call themselves hackers. They are not TV celebrities with wide name recognition, but everyone knows their achievements, which form a large part of our new, emerging society's technological basis: the Internet and the Web (which together can be called the Net), the personal computer, and an important portion of the software used for running them. The hackers' "jargon file," compiled collectively on the Net, defines them as people who "program enthusiastically" and who believe that "information-sharing is a powerful positive good, and that it is an ethical duty of hackers to share their expertise by writing free software and facilitating access to information and to computing resources wherever possible." This has been the hacker ethic ever since a group of MIT's passionate programmers started calling themselves hackers in the early sixties. (Later, in the mid-eighties, the media started applying the term to computer criminals. In order to avoid the confusion with virus writers and intruders into information systems, hackers began calling these destructive computer users crackers. In this book, this distinction between hackers and crackers is observed.)

My own initial interest in these hackers was technological, related to the impressive fact that the best-known symbols of our time - the Net, the personal computer, and software such as the Linux operating system --- were actually developed not by enterprises or governments but were created primarily by some enthusiastic individuals who just started to realize their ideas with other like-minded individuals working in a free rhythm. (Those who are interested in the details of their development may turn to the appendix, "A Brief History of Computer Hackerism".) I wanted to understand the internal logic of this activity, its driving forces. However, the more I thought about computer hackers, the more obvious it became that what was even more interesting about them, in human terms, was the fact that these hackers represented a much larger spiritual challenge to our time. Computer hackers themselves have always admitted this wider applicability of their ways. Their "jargon file" emphasizes that a hacker is basically "an expert or enthusiast of any kind. One might be an astronomy hacker, for example." In this sense, a person can be a hacker without having anything to do with computers.

The main question transformed into, What if we look at hackers from a wider perspective? What does their challenge then mean? Looking at the hacker ethic in this way, it becomes a name for a general passionate relationship to work that is developing in our information age. From this perspective, the hacker ethic is a new work ethic that challenges the attitude to work that has held us in its thrall so long, the Protestant work ethic, as explicated in Max Weber's classic The Protestant Ethic and the Spirit of Capitalism (1904-1905).

To some computer hackers, this kind of linking of the hacker ethic to Weber may at first seem alien. They should keep in mind that in this book the expression hacker ethic is used in a sense that extends beyond computer hackerism, and that for this reason it confronts social forces that are not normally considered in discussions concerned exclusively with computers. This expansion of the hacker ethic thus presents an intellectual challenge to computer hackers, as well.

But first and foremost the hacker ethic is a challenge to our society and to each of our lives. Besides the work ethic, the second important level of this challenge is the hacker money ethic - a level that Weber defined as the other main component of the Protestant work ethic. Clearly, the "information-sharing" mentioned in the hacker-ethic definition cited above is not the dominant way of making money in our time; on the contrary, money is mostly made by information-owning. Neither is the first hackers' ethos --- that activity should be motivated primarily not by money but rather a desire to create something that one's peer community would find valuable --- a common attitude. While we cannot claim that all present computer hackers share this money ethic or that it is likely to spread into society at large, as we can about their work ethic, we can say that it has been an important force in the formation of our time and that the hackers' debate over the nature of the information economy could lead to consequences at least as radical as those of their work ethic.

The third element present within the hacker ethic from the very beginning, touched upon in the cited definition by the phrase "facilitating access to information and to computing resources," could be called their network ethic or nethic. It has addressed ideas such as freedom of expression on the Net and access to the Net for all. Most computer hackers support only some parts of this nethic, but in terms of their social significance they must be understood as a whole. The impact of these themes remains to be seen, but they definitely go to the heart of the ethical challenges of the information age.

This book is based on an ongoing collaboration between its three authors, a collaboration taking place in various forms over several years (with Manuel Castells through research we conduct together in California, and with Linus Torvalds in the midst of just having fun). The idea for a book dealing with the hacker ethic was born the first time all three of us met, in the fall of 1998, when we were invited speakers at a symposium hosted by the University of California at Berkeley, that traditional hacker stronghold. At that time, we decided to expand our presentations, which dealt with the same subjects as the present work. Linus, we decided, would start as a representative of computer hackerism, Manuel would present his theory of our information age (consisting of the rise of informationalism, the new information-technology paradigm, and a new social form, the network society), and I would examine the social meaning of the hacker ethic by placing the example of Linus's computer hackerism against Manuel's larger background picture of our time. Naturally, each one of us would still speak for himself.

The book adheres to this plan: in his Prologue, "What Makes Hackers Tick? a.k.a. Linus's Law," Linus --- as the originator of one of the most famous hacker creations of our time, the Linux operating system - describes his view of the forces that contribute to the success of hackerism. Manuel has spent the last fifteen years on a study of our time, culminating in his three-volume, 1,500-page work, The Information Age (second

revised edition, 2000). In this book's Epilogue, "Informationalism and the Network Society," he presents for the first time the findings of his research, with some new important additions, in a form accessible to the general reader. My analysis is placed between Linus's and Manuel's and is divided in three parts accordingt to the three levels of the hacker ethic: the work ethic, the money ethic, and the nethic. (Some further elaborations of these themes can be found at the book's Website, www.hackerethic.org.)

Those readers who prefer to have a description of the theory background before, and not as a closing systematization of, my examination, may consult Manuel's epilogue right away. Otherwise, let Linus start.

***From The Hacker Ethic and the Spirit of the Information Age by Pekka Himanen with Linus Torvalds and Manuel Castells (Random House, 2001). For more, see www.hackerethic.org. This writing can be published freely on the web with this information included.***

# The Hacker Work Ethic

*This is an excerpt from Pekka Himanen's book* 'The Hacker Ethic'.

Linus Torvalds says in his Prologue that, for the hacker, "the computer itself is entertainment," meaning that the hacker programs because he finds programming intrinsically interesting, exciting, and joyous.

The spirit behind other hackers' creations is very similar to this. Torvalds is not alone in describing his work with statements like "Linux hackers do something because they find it to be very interesting." For example, Vinton Cerf, who is somtimes called "the father of the Internet," comments on the fascination programming exerts: "There was something amazingly enticing about programming." Steve Wozniak, the person who built the first real personal computer, says forthrightly about his discovery of the wonders of programming: "It was just the most intriguing world." This is a general spirit: hackers program because programming challenges are of intrinsic interest to them. Problems related to programming arouse genuine curiosity in the hacker and make him eager to learn more.

The hacker is also enthusiastic about this interesting thing; it energizes him. From the MIT of the sixties onward, the classic hacker has emerged from sleep in the early afternoon to start programming with enthusiasm and has continued his efforts, deeply immersed in coding, into the wee hours of the morning. A good example of this is the way sixteen-year-old Irish hacker Sarah Flannery describes her work on the so-called Cayley-Purser encryption algorithm, "I had a great feeling of excitement. . . . I worked constantly for whole days on end, and it was exhilarating. There were times when I never wanted to stop."

Hacker activity is also joyful. It often has its roots in playful explorations. Torvalds has described, in messages on the Net, how Linux began to expand from small experiments with the computer he had just acquired. In the same messages, he has explained his motivation for developing Linux by simply stating that "it was/is fun working on it." Tim Berners-Lee, the man behind the Web, also describes how this creation began with experiments in linking what he called "play programs." Wozniak relates how many characteristics of the Apple computer "came from a game, and the fun features that were built in were only to do one pet project, which was to program . . . [a game called] Breakout and show it off at the club." Flannery comments on how her work on the development of encryption technology evolved in the alternation between library study of theorems and the practice of exploratory programming: "With a particularly interesting theorem . . . I'd write a program to generate examples. . . . Whenever I programmed something I'd end up playing around for hours rather than getting back to plodding my way through the paper."

Sometimes this joyfulness shows in the hacker's "flesh life" as well. For example, Sandy Lerner is known not only for being one of the hackers behind the Internet routers but also for riding naked on horseback. Richar Stallman, the bearded and longhaired guru, attends computer gatherings in a robe, and he exorcises commercial programs from the

machines brought to him by his followers. Eric Raymond, a well-known defender of hacker culture, is also known for his playful lifestyle: a fan of live role-playing games, he roams the streets of his Pennsylvania hometown and the surrounding woods attired as an ancient sage, a Roman senator, or a seventeenth-century cavalier.

Raymond has also given a good summary of the general hacker spirit in his description of the Unix hackers' philosophy:

To do the Unix philosophy right, you have to be loyal to excellence. You have to believe that software is a craft worth all the intelligence and passion you can muster. . . . Software design and implementation should be a joyous art, and a kind of high-level play. If this attitude seems preposterous or vaguely embarrassing to you, stop and think; ask yourself what you've forgotten. Why do you design software instead of doing something else to make money or pass the time? You must have thought software was worthy of your passions once...

To do the Unix philosophy right, you need to have (or recover) that attitude. You need to care. You need to play. You need to be willing to explore.

In summing up hacker activity's spirit, Raymond uses the word passion, which corresponds to Torvalds's entertainment, as he defined it in the Prologue. But Raymond's term is perhaps even more apt because, even though both words have associations that are not meant in this context, passion conveys more intuitively than entertainment the three levels described above-the dedication to an activity that is intrinsically interesting, inspiring, and joyous.

This passionate relationship to work is not an attitude found only among computer hackers. For example, the academic world can be seen as its much older predecessor. The researcher's passionate intellectual inquiry received similar expression nearly 2,500 years ago when Plato, founder of the first academy, said of philosophy, "like light flashing forth when a fire is kindled, it is born in the soul and straightway nourishes itself."

The same attitude may also be found in many other spheres of life-among artists, artisans, and the "information professionals," from managares and engineers to media workers and designers, for example. It is not only the hackers' "jargon file" that emphasizes this general idea of being a hacker. At the first Hacker Conference in San Francisco in 1984, Burrell Smith, the hacker behind Apple's Macintosh computer, defined the term as follows: "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." Raymond notes in his guide "How to Become a Hacker" that "there are people who apply the hacker attitude to other things [than software], like electronics and music-actually, you can find it at the highest levels of any science or art."

Looked at on this level, computer hackers can be understood as an excellent example of a more general work ethic-which we can give the name the hacker work ethic-gaining ground in our network society, in which the role of information professionals is

expanding. But although we use a label coined by computer hackers to express this attitude, it is important to note that we could talk about it even without any reference to computer people. We are discussing a general social challenge that calls into question the Protestant work ethic that has long governed our lives and still maintains a powerful hold on us.

Let's see what type of long historical and strong societal forces the hacker work ethic, in this sense, faces. The familiar expression "Protestant work ethic" derives, of course, from Max Weber's famous essay The Protestant Ethic and the Spirit of Capitalism (1904-1905). Weber starts out by describing how the notion of work as a duty lies at the core of the capitalist spirit that arose in the sixteenth century: "This peculiar idea, so familiar to us to-day, but in reality so little a matter of course, of one's duty in a calling, is what is most characteristic of the social ethic of capitalistic culture, and is in a sense the fundamental basis of it. It is an obligation which the individual is supposed to feel and does feel towards the content of his professional activity, no matter in what it consists, in particular no matter whether it appears on the surface as a utilization of his personal powers, or only of his material possessions (as capital)." Weber goes on to say: "Not only is a developed sense of responsibility absolutely indispensable, but in general also an attitude which, at least during working hours, is freed from continual calculations of how the customary wage may be earned with a maximum of comfort and a minimum of exertion. Labour must, on the contrary, be performed as if it were an absolute end in itself, a calling."

Then Weber demonstrates how the other main force described in his essay, the work ethic taught by Protestants, which also arose in the sixteenth century, furthered these goals. The Protestant preacher Richard Baxter expressed that work ethic in its pure form: "It is for action that God maintaineth us and our activities; work is the moral as well as the natural end of power," and to say "I will pray and meditate [instead of working], is as if your servant should refuse his greatest work and tie himself to some lesser, easier part." God is not pleased to see people just meditating and praying --- he wants them to do their job.

True to the capitalist spirit, Baxter advises employers to reinforce this idea in workers of wanting to do one's job as well as possible by making it a matter of conscience: "A truly godly servant will do all your service in obedience to God, as if God Himself had bid him do it." Baxter sums up this attitude by referring to labor as a "calling," a good expression of the three core attitudes of the Protestant work ethic: work must be seen as an end in itself, at work one must do one's part as well as possible, and work must be regarded as a duty, which must be done because it must be done.

While the hacker work ethic's precursor is in the academy, Weber says that the Protestant ethic's only historical precursor is in the monastery. And certainly, if we expand on Weber's comparison, we can see many similarities. In the sixth century, for example, Benedict's monastic rule required all monks to see the work assigned to them as their duty and warned work-shy brethren by noting that "idleness is the enemy of the soul." Monks were also not supposed to question the jobs they were given. Benedict's fifth-century predecessor John Cassian made this clear in his monastic rule by describing in admiring tones the obedience of a monk, named John, to his elder's order to roll a stone so large that no human being could move it:

Again, when some others were anxious to be edified by the example of his [John's] obedience, the elder called him and said: "John, run and roll that stone hither as quickly as possible;" and he forthwith, applying now his neck, and now his whole body, tried with all his might and main to roll an enormous stone which a great crowd of men would not be able to move, so that not only were his clothes saturated with sweat from his limbs, but the stone itself was wetted by his neck; in this too never weighing the impossibility of the command and deed, out of reverence for the old man and the unfeigned simplicity of his service, as he believed implicitly that the old man could not command him to do anything vain or without reason.

This Sisyphean straining epitomizes the idea, central to monastic thought, that one should not question the nature of one's work. Benedict's monastic rule even explained that the nature of the work did not matter because the highest purpose of work was not actually to get something done but to humble the worker's soul by making him do whatever is told-a principle that seems to be still active in a great number of offices. In the medieval time, this prototype for the Protestant work ethic existed only within the monasteries, and it did not influence the prevailing attitude of the church, much less that of society at large. It was only the Protestant reformation that allowed the spread of monastic thinking to the world beyond the monastery walls.

However, Weber went on to emphasize that even though the spirit of capitalism found its essentially religious justification in the Protestant ethic, the latter soon emancipated itself from religion and began to operate according to its own laws. To use Weber's famous metaphor, it turned into a religiously neutral iron cage. This is an essential qualification. In our globalizing world, we should think of the term Protestant ethic in the same way we think of an expression such as platonic love. When we say that someone loves another person platonically, we do not mean that he is a Platonist-that is, an adherent of Plato's philosophy, metaphysics and all. We may attribute a platonic love relationship to a follower of any philosophy, religion, or culture. In the same way, we can speak of someone's "Protestant ethic" regardless of his or her faith or culture. Thus, a Japanese person, an atheist, or a devout Catholic may act-and often does act-in accordance with a Protestant ethic.

One need not look very far to realize how strong a force this Protestant ethic still is. Commonplace remarks like "I want to do my job well," or those made by employers in their little speeches at employee retirement parties about how a person "has always been an industrious/responsible/reliable/loyal worker" are the legacy of the Protestant ethic in that they make no demands on the nature of the work itself. The elevation of work to the status of the most important thing in life-at its extreme, a work addiction that leads to complete neglect of one's loved ones-is another symptom of the Protestant ethic. So is work done with clenched jaws and a responsibility-ridden attitude and the bad conscience many feel when they have to miss work due to ill health.

Seen in a larger historical context, this continued dominance of the Protestant ethic is not so surprising when we remember that even though our network society differs in many significant ways from its predecessor, the industrial society, its "new economy" does not involve a total break with the capitalism Weber describes: it is merely a new kind of capitalism. In The Information Age, Castells stresses that work, in the sense of labor, is not about to end, despite wild paradisiacal forecasts such as Jeremy Rifkin's The End of Work. We easily fall for this illusion that technological advances will,

somehow, automatically, make our lives less work-centered-but if we just look at the statistical facts of the rise of the network society so far and project them into the future, we must agree with Castells on the nature of the prevailing pattern: "Work is, and will be for the foreseeable future, the nucleus of people's life." The network society itself does not question the Protestant ethic. Left to its own devices, the work-centered spirit easily continues to dominate within it.

Seen in this overall context, the radical nature of hackerism consists of its proposing an alternative spirit for the network society-a spirit that finally questions the dominant Protestant ethic. In this context, we find the only sense in which hackers are really crackers: they are trying to crack the lock of the iron cage.

**The Purpose of Life**

The displacement of the Protestant ethic will not happen overnight. It will take time, like all great cultural changes. The Protestant ethic is so deeply embedded in our present consciousness that it is often thought of as if it were just "human nature." Of course, it is not. Even a brief look at pre-Protestant attitudes toward work provides a healthy reminder of that fact. Both the Protestant and the hacker ethic are historically singular.

Richard Baxter's view of work was completely alien to the pre-Protestant church. Before the Reformation, clerics tended to devote time to questions such as "Is there life after death?" but none of them worried about whether there was work after life. Work did not belong among the church's highest ideals. God himself worked for six days and finally rested on the seventh. This was the highest goal for human beings as well: in Heaven, just as on Sundays, people would not have to work. Paradise was in, office was out. One might say that Christianity's original answer to the question "What is the purpose of life?" was: the purpose of life is Sunday.

This statement is not just a witticism. In the fifth century, Augustine compared our life quite literally to Friday, the day when, according to the teachings of the church, Adam and Eve sinned and Christ suffered on the cross. Augustine wrote that in Heaven we'll find a perennial Sunday, the day on which God rested and Christ ascended to Heaven: "That will truly be the greatest of Sabbaths; a Sabbath that has no evening." Life is just a long wait for the weekend.

Because the Church Fathers saw work as merely a consequence of the fall from grace, they also took very particular conceptual care in their descriptions of Adam's and Eve's activities in Paradise. Whatever Adam and Eve may have done there, it could not be seen as work. Augustine emphasizes that in Eden "praiseworthy work was not toilsome"-it was no more than a pleasant hobby.

The pre-Protestant churchmen understood work, "toil," as punishment. In medieval visionary literature that speaks to churchmen's images of Hell, the implements of labor fully reveal their true nature as instruments of torture: sinners are punished with hammers and other tools. What's more, according to these visions, there is in Hell an even more cruel torture than the directly inflicted physical one: perennial toil. When the devout brother Brendan saw, in the sixth century, a worker on his visit to the beyond, he immediately made the sign of the cross: he realized that he had arrived where all hope must be abandoned. Here is the narrator of his vision:

When they had passed on further, about a stone's throw, they heard the noise of bellows blowing like thunder, and the beating of sledge hammers on the anvils and iron. Then St. Brendan armed himself all over his body with the sign of the Cross, saying, "O Lord Jesus Christ, deliver us from this sinister island." Soon one of the inhabitants appeared to do some work. He was hairy and hideous, blackened with fire and smoke. When he saw the servants of Christ near the island, he withdrew into his forge, crying aloud: "Woe! Woe! Woe!"

If you do not conduct yourself well in this life, the thinking went, you are condemned to work even in the next. And, even worse, that work, according to the pre-Protestant church, will be absolutely useless, meaningless to an extent you could never have imagined even on your worst working day on earth.

This theme crystallizes in the apotheosis of the pre-Protestant worldview, Dante's Divine Comedy (completed just before his death in 1321), in which sinners who have devoted their lives to money-both spendthrifts and misers-are doomed to push huge boulders around an eternal circle:

```
More shades were here than anywhere above,
and from both sides, to the sounds of their screams,
straining their chests, they rolled enormous weights.
:
And when they met and clashed against each other
they turned to push the other way, one side
screaming, "Why hoard?", the other side, "Why waste?"
:
And so they moved back round the gloomy circle,
returning on both sides to opposite poles
to scream their shameful tune another time;
:
again they came to clash and turn and roll
forever in their semicircle joust.
```

Dante borrows this idea from Greek mythology. In Tartarus, where the very worst human beings were dispatched, the most severe punishment was meted out to greedy Sisyphus, who was doomed to endlessly push a big rock up to the top of a hill, from which it always rolled back down. Sunday always beckons to Sisyphus and the sinners in Dante's Inferno, but it never comes. They are condemned to an eternal Friday.

Considering this background, we can now gain a better understanding of how great a change in our attitude to work the Protestant Reformation entailed. In allegorical terms, it moved life's center of gravity from Sunday to Friday. The Protestant ethic reoriented ideology so thoroughly that it even turned Heaven and Hell upside down. When work became an end in itself on earth, the clerics found it difficult to imagine Heaven as a place for mere time-wasting leisure, and work could no longer be seen as infernal punishment. Thus, reformed eighteenth-century cleric Johann Kasper Lavater explained that even in Heaven "we cannot be blessed without having occupations. To have an occupation means to have a calling, an office, a special, particular task to do." Baptist William Clarke Ulyat put it in a nutshell when he described Heaven at the beginning of the twentieth century: "practically it is a workshop."

The Protestant ethic proved so powerful that its work-centeredness permeated even our imagination. A great example of this is Daniel Defoe's Robinson Crusoe (1719), a novel written by a man trained as a Protestant preacher. Marooned on an abundant island, Crusoe does not take it easy; he works all the time. He is such an orthodox Protestant that he does not even take Sunday off, though he otherwise still observes the seven-day week. After saving an aborigine from his enemies, he aptly names him Friday, trains him in the Protestant ethic, and then praises him in a manner that perfectly describes that ethic's ideal worker: "Never man had a more faithful, loving, sincere servant, perfectly obliged and engaged; his very affections were ty'd to me, like those of a child to a father."

In Michel Tournier's twentieth-century satirical retelling of the novel, Vendredi (Friday), Friday's conversion to the Protestant ethic is still more total. Crusoe decides to put Friday to the test by giving him a task even more Sisyphean than what Cassian's monastic rule prescribed:

I set him a task which in every prison in the world is held to be the most degrading of harassments-the task of digging a hole and filling it in with the contents of a second; then digging a third, and so on. He labored at this for an entire day, under a leaden sky and in heat like that of a furnace. . . . To say that Friday gave no sign of resenting this idiotic employment, is not enough. I have seldom seen him work with such good will.

Sisyphus has truly become a hero.

**The Passionate Life**

When the hacker ethic is placed in this large historical context, it is easy to see that the hacker ethic --- understood not just as the computer hackers' ethic but as a general social challenge --- resembles the pre-Protestant ethic to a much greater degree than it does the Protestant one. In this sense, one could say that for hackers the purpose of life is closer to Sunday than to Friday. But, it is important to note, only closer: ultimately, the hacker ethic is not the same as the pre-Protestant work ethic, which envisions an attainable paradise of life without doing anything. Hackers want to realize their passions, and they are ready to accept that the pursuit even of interesting tasks may not always be unmitigated bliss.

For hackers, passion describes the general tenor of their activity, though its fulfillment may not be sheer joyful play in all its aspects. Thus, Linus Torvalds has described his work on Linux as a combination of enjoyable hobby and serious work: "Linux has very much been a hobby (but a serious one: the best type)." Passionate and creative, hacking also entails hard work. Raymond says in his guide "How to Become a Hacker," "Being a hacker is lots of fun, but it's a kind of fun that takes a lot of effort." Such effort is needed in the creation of anything even just a little bit greater. If need be, hackers are also ready for the less interesting parts necessary for the creation of the whole. However, the meaningfulness of the whole gives even its more boring aspects a worth. Raymond writes: "The hard work and dedication will become a kind of intense play rather than drudgery."

There's a difference between being permanently joyless and having found a passion in life for the realization of which one is also willing to take on the less joyful but nonetheless necessary parts.

***From* The Hacker Ethic and the Spirit of the Information Age *by Pekka Himanen with Linus Torvalds and Manuel Castells (Random House, 2001). For more, see www.hackerethic.org. This writing can be published freely on the web with this information included.***

# The Academy and the Monastery

*This is an excerpt from Pekka Himanen's book* 'The Hacker Ethic'.

Dedicated to Eric Raymond - the introducer of the allegory of the cathedral and the bazaar

## The Open Model

According to the hackers' "jargon file," the original hacker ethic meant the belief that "information-sharing is a powerful positive good." In practice this meant the ethical duty to work through an open development model, in which the hacker gives his or her creation freely for others to use, test, and develop further. Although for the author of this writing, the ethical arguments for the hacker model 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 arguments for the hackers' passionate and free way of working 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 of taking a closer look at the hackers' idea of openness from this purely pragmatic viewpoint. Thus all observations in this essay will be purely pragmatic; those who want to read more about the ethical arguments for the hacker model may turn to my book The Hacker Ethic and the Spirit of the Information Age with Linus Torvalds and Manuel Castells (Random House, 2001).

The development of the Net is an excellent concrete example of the hacker ethic in action, but the Linux project, which has arguably taken the ideal of openness the furthest so far, serves as an even better one. Torvalds started working on Linux in 1991 while he was a student at the University of Helsinki. 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. By studying and using it as a developmental framework, he proceeded to design his own operating system. 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." 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.

The next, improved version was available as soon as early October. Torvalds then extended an even more direct invitation to hackers to join him in the development of the new system. 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.

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. 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 authority 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 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 his well-known essay "The Cathedral and the Bazaar," published originally on the Net, Eric 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.

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

Although Raymond's allegory of the bazaar and the cathedral elegantly captures the difference between the open-source and closed-source models, I would like to explain the power of the open model vis-a-vis the closed model further by suggesting another pair of allegories: the academy and the monastery. In fact, the open-source model resembles the academy even more directly than the bazaar. 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 sociologist Robert Merton wrote that this idea of self-correction was as important a principle to science as openness. He called it organized skepticism - historically, it is a continuation of the synusia of Plato's Academy, which also included the idea of approaching the truth through critical dialogue. 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.

Of course, scientists 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 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, too-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 = mc2"-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 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 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 versions"). 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. In the broadest sense, there have been only three long-lived research paradigms: 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 will have to accept the result purely as it is. Other uses of it are called "unauthorized uses." We can 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 or make curious inquiries about what is being done." The closed model does not allow for initiative or criticism that would enable an activity to become more creative and self-corrective.

It is true that many hackers oppose hierarchical operation for ethical reasons, like 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 = mc2 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 their model is so powerful. Hackers and scientists can just begin 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 has been turned into a governmental bureaucracy or monastery.)

But the relative lack of structures does not mean that there are no structures. Despite its appearance, hackerism does not exist in a state of anarchy or mean paradisiacal utopianism 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 to support 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 to a large extent 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. 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. Most hackers have learned programming in a similar informal way, following their passions. The examples of the ability of ten-year-olds to learn very complicated programming issues tell us much about the importance of passion in the learning process, as opposed to the slow going their contemporaries often find their education in traditional schools to be.

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 tested the features of the processor by writing out either As or Bs gradually expanded into a plan for a Net newsgroup-reading program and then on to the ambitious idea of an entire operating system. 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 typically 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 neophytes think. For an expert, empathizing with someone who is just learning something involves levels of simplification that he or she often resists for personal 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 companions in learning (synetheis). 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, a matchmaker, and a master of ceremonies at banquets. 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, we can think of the best experts as the community's gadflies, midwives, and symposiarchs. They are able to entice the collective learning process, which links the learning and development models together as a powerful hacker way of creating things.

***From* The Hacker Ethic and the Spirit of the Information Age *by Pekka Himanen with Linus Torvalds and Manuel Castells (Random House, 2001). For more, see www.hackerethic.org. This writing can be published freely on the web with this information included.***