

Prologue
LINUS TORVALDS

Epilogue
MANUEL CASTELLS



R A N D O M H O U S E / N E W Y O R K

The
HACKER
Ethic

and the Spirit of
the Information Age

PEKKA HIMANEN

to all the existing systems. The hacker community is not unified in its answers to these big questions, but even just having started a debate on these questions in the nucleus of the information economy is a radical enough challenge.

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 *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.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 Tannenbaum'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 (*synthesis*).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 officeholder 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."

11. Brand, *Media Lab*, p. 57.
12. *The Importance of Living*, p. 158. He adds: "The danger is that we get over-civilized and that we come to a point, as indeed we have already done, when the work of getting food is so strenuous that we lose our appetite for food in the process of getting it."
13. Linzmayer, *Apple Confidential* (1999), pp. 37–40.
14. Wolfson and Leyba, "Humble Hero."
15. Southwick, *High Noon: The Inside Story of Scott McNealy and the Rise of SUN Microsystems* (1999), p. 16. For the history of the company's founding, see chap. 1.
16. Ceruzzi, *A History of Modern Computing* (1998), chap. 7. Among Microsoft's first languages were BASIC (1975), FORTRAN (1977), and COBOL-80 (1978). From the later perspective of Microsoft's attacks on Unix-like operating systems (the latest case being the attacks on Linux in the internal memorandums that were leaked to the public: Valloppillil, *Open Source Software* [1998]; Valloppillil and Cohen, *Linux OS Competitive Analysis* [1998]), it is a bit ironic that its first operating system was also a version of the hacker-favored Unix-family, XENIX ("Microsoft Timeline").
17. Gates, *The New York Times Syndicate*.
18. The story of Red Hat is described in Young with Goldman Rohm, *Under the Radar* (1999).
19. Stallman, "The Free Software Song."
20. "What Is Free Software?" (1996). For other serious treatments of the topic, see "The GNU Manifesto" (1985) and "The GNU Operating System and the Free Software Movement" (1999).
21. This is the difference between those who prefer Stallman's *free software* and those who prefer *open source*. One of the reasons for adopting the new term suggested by Chris Peterson in a meeting of a few leading hackers in Palo Alto in February 1998 was to be less ideological. The two most famous proponents of this new term are Bruce Perens and Eric Raymond, who founded opensource.org for spreading the idea. Cf. Opensource.org, "History of the Open Source Initiative." Cf. also Rosenberg, *Open Source: The Unauthorized White Papers* (2000) and Wayner, *Free for All: How Linux and the Free Software Movement Undercut the High-Tech Titans* (2000).
22. Anthony, *The Ideology of Work* (1977), p. 92.

Chapter 4: The Academy and the Monastery

1. Weber, *Protestant Ethic*, p. 64.
2. *The Jargon File*, s.v. *hacker ethic*.
3. For the early history, see Torvalds, "Re: Writing an OS" (1992) and "Birthday" (1992).
4. Cf. Tanenbaum, *Operating Systems: Design and Implementation* (1987).
5. Torvalds, "What Would You Like to See Most in Minix?" (1991).
6. Torvalds, "Birthday" (1992).
7. On October 5, 1991, Torvalds posted a message asking, "Do you pine for the nice days of minix-1.1, when men were men and wrote their own device drivers?" Torvalds, "Free Minix-like Kernel Source for 386-AT" (1991).
8. For a more comprehensive look at the contributors to the Linux project, see Torvalds, "Credits," and Dempsey, Weiss, Jones, and Greenberg, *A Quantitative Profile of a Community of Open Source Linux Developers* (1999).
9. The first discussion took place within the newsgroup comp.os.minix. Linux 0.0.1 became available on the Finnish server nic.funet.fi in the directory /pub/OS/Linux in September 1991. Nowadays, Torvalds uploads the newest version of the kernel to ftp.kernel.org/pub/linux/kernel. There are countless mailing lists, newsgroups, and webpages focused on Linux.
10. Raymond writes:

The most important feature of Linux, however, was not technical but sociological. Until the Linux development, everyone believed that any software as complex as an operating system had to be developed in a carefully coordinated way by a relatively small, tightly knit group of people. This model was and still is typical of both commercial software and the great free-ware cathedrals built by the Free Software Foundation in the 1980s—also of the freeBSD/netBSD/OpenBSD projects that spun off from the Jolitzes' original 386BSD port.

Linux evolved in a completely different way. From nearly the beginning, it was rather casually hacked on by huge numbers of volunteers coordinating only through the Internet. Quality was maintained not by rigid standards or autocracy but by the naively simple strategy of releasing every week and

getting feedback from hundreds of users within days, creating a sort of rapid Darwinian selection on the mutations introduced by developers. ("The Cathedral and the Bazaar" [1999], pp. 23–24)

11. Merton, "Normative Structure of Science," in *Sociology of Science*. Cf. p. 277.
12. Basically, all of Plato's Socratic dialogues are examples of this critical dialogue; in them, Socrates often makes remarks about the need for critical dialogue. For example, in *Crito*, Socrates says, "Let us examine the question together, my dear friend, and if you make any objection while I am speaking, make it and I will listen to you" (48e). In *Phaedo*, he incites his interlocutor to criticize him by asking, "Do you think there is something lacking in my argument?" and in *Euthydemus* he remarks similarly: "There is nothing I would like better than to be refuted on these points" (295a). In *Theaetetus* and *Clitophon*, Socrates explains why the process of critique is always beneficial: "Either we shall find what we are going out after, or we shall be less inclined to think we know things which we don't know at all—and even that would be a reward we could not fairly be dissatisfied with" (187b–c); and, "Once I know my good and bad points, I will make it my practice to pursue and develop the former while ridding myself of the latter to the extent that I am able" (407a). For this reason, in academic discussion one should present a critique frankly and not try to please anyone (cf. *Euthyphro*, 14e; *Protagoras*, 319b, 336c; *Republic*, 336e).
13. In fact, William Whewell, who coined the word *scientist* in the nineteenth century, meant by that word a person participating in such a self-corrective process.
14. Kuhn said that paradigms are "universally recognized scientific achievements that for a time provide model problems and solutions to a community of practitioners" (*The Structure of Scientific Revolutions* [1962], p. x).
15. Basil, *Long Rules*, 48.
16. For Torvalds's description of his first programming experiments, such as a submarine game, see Learmonth, "Giving It All Away" (1997). Similarly, Wozniak became excited by technology in the fourth grade, and in the sixth he built a computer that played tic-tac-toe. Wozniak describes how his learning proceeded: "It was all self-done; I didn't even take a course, didn't ever buy a book on how

to do it" (Wolfson and Leyba, "Humble Hero"). In another context, he adds: "It's much more important to get a student motivated and want to learn something . . . than it is just strictly to teach it, teach it, teach it and expect that it gets absorbed" (Tech, "An Interview with Steve Wozniak" [1998]).

17. Torvalds, "Re: Writing an OS" (1992).
18. Cf. Plato, *Minos*, 319e.
19. Plato describes the idea of midwifery through the mouth of Socrates, whom he has say in one of his dialogues:

one thing which I have in common with the ordinary midwives is that I myself am barren of wisdom. The common reproach against me is that I am always asking questions of other people but never express my own views about anything, because there is no wisdom in me; and that is true enough. And the reason of it is this, that God compels me to attend the travail of others, but has forbidden me to procreate. So that I am not in any sense a wise man; I cannot claim as the child of my own soul any discovery worth the name of wisdom. But with those who associate with me it is different. At first some of them may give the impression of being ignorant and stupid; but as time goes on and our association continues, all whom God permits are seen to make progress—a progress which is amazing both to other people and to themselves. And yet it is clear that this is not due to anything they have learned from me; it is that they discover within themselves a multitude of beautiful things, which they bring forth into the light. (Plato, *Theaetetus*, 150c–d)

Plutarch sums up: "Socrates was not engaged in teaching anything, but by exciting perplexities as if inducing the inception of labour-pains in young men he would arouse and quicken and help to deliver their innate conceptions; and his name for this was obstetric skill, since it does not, as other men pretend to do, implant in those who come upon it intelligence from without but shows that they have it native within themselves but undeveloped and confused and in need of nurture and stabilization" (*Platonic Questions*, 1000e).

The Socratic idea is that the purpose of teaching is to help someone learn to learn, to be able to pose questions. A precondition for that is puzzlement. In the dialogue *Meno*, the title character describes the Socratic teacher's effect:

Socrates, before I even met you I used to hear that you are always in a state of perplexity and that you bring others to the same state, and now I think you are bewitching and beguiling me, simply putting me under a spell, so that I am quite perplexed. Indeed, if a joke is in order, you seem, in appearance and in every other way, to be like the electric ray, for it too makes anyone who comes close and touches it feel numb, and you now seem to have had that kind of effect on me, for both my mind and my tongue are numb, and I have no answer to give you. (80a–b)

But this state of perplexity is ultimately for the better, as Socrates explains:

SOCRATES: Have we done him any harm by making him perplexed and numb as the torpedo fish does?

MENO: I do not think so.

SOCRATES: Indeed, we have probably achieved something relevant to finding out how matters stand, for now, as he does not know, he would be glad to find out, whereas before he thought he could easily make many fine speeches to large audiences about the square of double size and said that it must have a base twice as long.

MENO: So it seems.

SOCRATES: Do you think that before he would have tried to find out that which he thought he knew though he did not, before he fell into perplexity and realized he did not know and longed to know? (84a–c; see also *Alcibiades*, 106d)

20. The reason the Socratic teacher was also called a matchmaker was that it was his task to join people into giving birth together (Xenophon, *Symposium*, 3). Socrates describes his method: “With the best will in the world I undertake the business of match-making; and I think I am good enough—God willing—at guessing with whom they might profitably keep company. Many of them I have given away to Prodicus; and a great number also to other wise and inspired persons” (Plato, *Theaetetus*, 151b). Compare this to: “Someone asked Aristippus [a disciple of Socrates] how Socrates had helped him. He replied, ‘He enabled me to find for myself satisfying fellow-students of philosophy’” (Philodemus, *Rhetoric*, 1, 342.13).

21. The third Academy metaphor was that of the teacher as master of ceremonies (the *symposiarkhos*) at banquets. These took place in the evenings, and in conjunction with the dialogues of the day they were an essential learning experience. The purpose of these banquets was quite serious and intellectually ambitious—for example, the discussion of some heavyweight philosophical subject—but they were, in addition, powerfully experiential events. (Two great descriptions are the symposia of Plato and Xenophon.)

The symposiarch was responsible for the success of the banquets in two ways: first, from his elevated position he made sure that the intellectual goals of dialogue were attained; second, it was also his responsibility to ensure that none of the participants remained too stiff. To this latter end, he had two means at his disposal. First, he had the right to order the excessively stiff participants to drink more wine. If this did not work, the symposiarch could order the participant to remove his clothes and dance! The symposiarch used any means necessary to catalyze passionate contributions (cf. Plato, *Symposium*, 213e–14a).

22. Plato, *Republic*, 7.536e.

23. *The Rule of St. Benedict*, 6.

24. Slowly, these themes are winning more space in theories of education. There is a renewed interest in collaborative learning, inspired largely by Vygotsky’s concept of the zone of proximal development, which stresses that an individual’s potential capability when cooperating with a more experienced person is greater than his or her actual capability in isolation (*Mind in Society* [1978]). When learners set questions themselves and work together, they can also learn from each other—they can benefit from the fact that there are always some learners who are more advanced. This is why Lave and Wenger find it important for learners and researchers to be in dialogue with one another. They speak of the novice’s “legitimate peripheral participation” in the expert culture (*Situated Learning: Legitimate Peripheral Participation* [1991]). The cautious formulation suggests what most university professors think about this idea.

Chapter 5: From Netiquette to a Nethic

1. The best expression of the netiquette that the hacker community shares is in “Netiquette Guidelines” by the Internet Engineering Task Force (RFC 1855), although it emphasizes that its purpose is not to “specify an Internet standard of any kind.” Another important