10

Science as Open Source Process

Friedrich Kittler
Translated by Peter Krapp

Ladies and Gentlemen,

If I had begun my fifteen minute intervention not in the European style with "Ladies and Gentlemen" but instead addressed you in C style with "hello world," then I would presumably not be able to count on my liberty for too long. I would have had the pleasure of spending my time under a compiler and an assembler instead of entering silly ASCII keystrokes, but who knows whether this wonderful ASCII sequence, *print f ("Hello world");* will not soon be protected by source copyright, even here in Europe.

So I have two concerns. First, I worry about carrying coals to Newcastle, because I could not be here earlier. And secondly, I worry that academic freedom will stand or fall with the freedom of source code. In referring to academia, I mean above all the university, without wishing to talk self-servingly *pro domo*. For it is crucial for the university, since the Athenians, that the knowledge generated and passed on by it must be able to circulate without the protection of patents and copyrights, unlike in closed or even secret research organizations and industries. I would like to elucidate this history briefly with an eye to the dangers that imperil academia today.

The European universities I refer to were a creation of the High Middle Ages, and as far as I can see without any models or predecessors in any other cultures. This uniqueness is based on a media-technological reason: knowledge proliferated not only in oral explications or lectures from docents to students who in turn might become teachers, but in contrast to the ancients Greeks, at this university one had to work, not just chat. With the introduction of paper, which was cheaper and lighter than papyrus, universities ran *scriptoria*, where lectures were transcribed and handwritten notes copied, as well as libraries, which archived these processed data. Thirdly, to make the parallel to the contemporary global net obvious, universities also had their own medium of transmission: as incredible as this sounds, they had their own courier services.

In early modernity, the universities lost this wondrous hardware of processing, storage and transmission which defines every computer. The developing territorial states and later nation states disallowed or swallowed the mail services of butchers, monasteries and universities, and deregulation, as you know, only happened recently. After Gutenberg's invention of the printing press, the lion's share of the knowledge produced in the university fell to the system of books and publishing

houses. Suddenly the universities ceased to write books and merely stored the books printed by others in libraries as well as in the heads of their students.

I leave open how Gutenberg's media revolution changed knowledge, modern universities, academies and labs. I am interested in this revolution above all in the context of open source and free software, as a rather precise model of what is going on these days. Without exaggerating too much, one may perhaps say that computers, at least on the software side, are also a creation of the university. If hardware, on its long march from tubes to transistors to integrated circuits, largely comes from military technology, the universal Turing machine as a concept (as software) stems from an academic dissertation, answering certain unsolved questions of the mathematical institute in Goettingen. Turing told Hilbert, if you like. Accordingly, the still dominant von-Neumann-architecture was developed by someone who made it from mathematics professor at Goettingen to chief consultant of the Pentagon. On its way to power, the knowledge contained in the computer and its algorithms has once again experienced the closure that threatened universities in their take-over by nation states.

As far as I know, the fastest and best algorithm testing prime numbers remains a trade secret of the Pentagon. The hope for pure, which is to say academic, mathematicians that pure math was never going to be abused for earthly ends, as Hardy still wrote during World War II, has been dramatically disappointed. But the parallel I draw between early modernity and the present only comes out fully in the so-called PC revolution. It was no accident that the garages and tinkerers' rooms that laid the groundwork for global firms like Intel and Apple were and are located next to or even on the grounds of institutions like the Rand Corporation or the Leland Stanford Junior University. The computer industry does what Gutenberg's printing press did when it took over and industrialized the calligraphy of the medieval university. The headhunters of Microsoft lurk around Stanford and at other doors of computer science departments, catch new programming serfs with new algorithms and squeeze them for five years, until the algorithms become proprietary and the coders, with their stock options, are dismissed into early retirement.

The worst aspect of this scandal seems to me that nobody talks about it. An American common law whose reach extends from the European Commission to the People's Republic of China has made an impossible concept of intellectual property as ubiquitous as it is unquestioned. Machines that, according to Turing's proof, are able to be not only all other machines but equally all human calculation, are now supposed to legitimate patents and copyrights more profoundly than ever. Machines that, according to the most recent results, run fastest and most efficiently when they were not programmed by programmers but by themselves, are supposed to belong to humans as private property—perhaps by way of euphemism for the capital corporate interests. Humanism, one might say, is today as in early modernity nothing but a fig leaf.

You all know better than me that the critique of this system can only be a practical one. Theoretical or historical remarks like mine can at best help not to lose one's overview among all the upgrades and benchmarks. However, it was practical when some programmers at the MIT resisted venality and when a computer science student at the University of Helsinki overcame the widespread fear of assemblers and cold starts. That is how immediately open source and free software are connected to the university. Look how much "edu" is in the sources of the Linux kernel. That is also how directly the future of the university depends on these free sources.

When the printing press and the nation state swallowed the media technologies of the medieval university, knowledge was left pretty much untouched on the content-level. The storage and transmission were privatized or nationalized, but data processing proper was still conducted in that beautiful old feedback circuit of eyes, ears, and writing hands. That is what changed with the computer revolution. Universal Turing machines make especially this data processing technologically reproducible. They see to it that the differences between the knowledge about technology, natural sciences, and humanities progressively disappear. This revolution, in other words, concerns all the faculties of the university, only to level their old distinctions that grew from media technologies.

The conflict of the faculties as Kant had described it may be solved peacefully, simply because it is no longer a matter of books against labs against counsel in the different faculties, but because all knowledge, including cultural knowledge, is processed in computers. This, it seems to me, grants an essential part of their chances to open source and free operating systems.

As usual, mathematicians may have been the first to grasp what this freedom delivers. Worldwide, two academic publishers distribute the mathematical journals of record. It should be obvious to catapult these journals from the Gutenberg galaxy into the Turing galaxy, especially after Adobe & Co successfully pirated almost the entire set of lead fonts from Garamond to Zapf. All mathematical knowledge would move to fully electronic publications, and their price as well as their copyright would be under the control of the said two global publishers. It is possible that this calculation goes awry, though: weeks or months before the essays or dissertations that advance mathematics land on an editor's desk in Heidelberg or Amsterdam, they are already on the computer server of a mathematical institute. This kind of bypass operation is more obvious than commercial distribution. The university can put its innovations online.

If this example, which I did not make up, is imitated, then the outlook for the commercialization of software is not rosy. The only way remaining to make knowledge proprietary would be to embed it in hardware. Once something is burned into a chip, it belongs to the firm who invested millions into the design and billions into their mass fabrication. No university can compete with that, regardless of whether it still depends on the financial support of a nation state or (more likely) already drifted off into medieval independence.

It seems as significant to me as it is sad that our congress deals with open source and free software, but does not even begin to discuss the possibility of open hardware. Since Gutenberg, constellations where, as in the middle ages, the hardware of knowledge resides with knowledge seem unthinkable. It is my impression that there are only two hopes left for hardware. Either academic freedom, while not building its own CPUs, can still produce a critique that would make faulty chips, or such stupid command sets as the complex instruction set of Intel, impossible. The division bug on the first two steps of the Pentium processor—A and B I believe—was discovered by a university, which forced Intel to conduct a recall that cost millions.

Or else, this chip- and hardware-production may act as its only possible critique. For if the prices for design and production of a machine that can be all other machines may climb to astronomical levels, they may also drop to zero. The first practical success of Turing's machine was that the Wehrmacht had forgotten an elementary fact: anything encoded by a machine can also be decoded by a machine. What Advanced Micro Devices calls reverse engineering is one of the best reasons why the mass market price of CPUs is now inexorably tending towards zero.

Free sources and open operating systems only have a chance because the computer industry always already undermines its own concept of property. Before Linux was ported to different hardware platforms, it was a highly specialized software that is said to have dismayed Andrew Tanenbaum (of Minix) by relying minimally on the Intel 386. Everything Linus Torvalds needed to that end was a publicly accessible programmer's manual, the software-abstraction of its hardware.

This may lead to a confident conclusion. "In the future," Bill Gates is supposed to have said in a perhaps not proprietary, but still internal, memo recently, "we will treat the end user as we treat computers: both are programmable." But as long as there are people who themselves are able to program instead of being programmed, this vision hopefully has no future.

Note

Originally delivered at "Wizards of OS," a conference organized by Volker Grassmuck and the Federal Office for Political Education [Bundeszentrale für Politische Bildung], House of World Cultures, Berlin, July 1999.