

The Anarchist Library
Anti-Copyright



The Two Economies

Kevin Carson

December 5, 2005

Joel Schlosberg sent me a transcript of Graham Seaman's excellent talk on "The Two Economies."

Seaman treats his model of the two economies as analogous to the early modern period, when the new economy emerged from the guild system:

So I say that's the first prerequisite — that you have people, that you have the old system actually producing the situation where there are people who need to get out of it, people who do not "fit" with the old system. Secondly, once these new people exist, they start infecting the old system...

Now I claim that there are bits of elements similar to that in the present as well. You have the old system becoming, the old system first generating the new one. It's becoming increasingly hard for the old system to produce software products. There are many products — especially ones that require cooperation of some kind, that require some kind of sharing, even commercially, that simply can't be produced under

Kevin Carson
The Two Economies
December 5, 2005

Retrieved on 3rd September 2021 from mutualist.blogspot.com

theanarchistlibrary.org

commercial constraints. And you see attempts at organization of standards bodies purely by software companies with one another, and the things break down. [some words are hard to hear] They are just inherently very, very bad at doing anything that requires real cooperation with one another. So you have some kinds of products, and I think this is software products and I think it's going to increase in the future, that will exist as free software that do not exist as proprietary software, and won't ever exist as proprietary software. Second..., I would say that as software industries become the leading sector of the economy, in many ways they're a sector which is not really producing profits; they're a sector which is taking profits from other parts of the economy... It's not, on the whole — commercial software, I do not believe, is on the whole, a large-scale creator of profits in itself. It's a reorganizer, redistributor of profits held by monopoly and the law of copyright...

Personally ... I think that the value of that software is the value of the CD that it's on, it's the value of the work that went into copying the CD, and so on; and is absolutely minimal. So, in my opinion, nearly all of the price of that comes purely from laws which allow monopolies over software, it comes from copyright laws and so on...

Okay, so you have a market; you have some elements of the old system being unable to actually continue making a profit in the leading sector of its economy without the help of quite repressive, and increasingly repressive laws, that as far as I can see are going to go on to become even more repressive and interfering in ever-larger areas of peoples' lives, that go well

beyond these non-material products themselves. You have people working completely outside that system producing products which become very difficult for the old system to produce and you have the old system being forced by standard economic reasons to take up the new products. You have free software working its way into the old economy, spreading throughout it at quite a high rate, not just people; partly this is because firms are, especially over the last couple of years, have been forced to reduce their IT costs, so there is a big temptation, especially for the bigger firms, to say “Well, why can’t we use free software instead of paying for the new Microsoft licensing system?” So that’s happening. You have companies that are starting to say, “How can we as a small company compete with the big software houses?” One way to do that is to use free software as a kind of tool for competition... But in doing so, they bring free software practices inside their own companies; they start to lose control. This gets to a point where, to some extent, managers just can no longer make arbitrary decisions about the form of software, about the contents, about the way it’s developed, because they can’t alienate the external people who work with them; they have to conform to their ideas, practices of free software developers. So you have the new system spreading back into the old, and starting to affect it.

There’s a lot of discussion of the specific ways that the new system is affecting the old, and a libertarian Marxist class analysis of the groups in the new economy with an interest in breaking through the restrictions of the old, that’s too complicated to summarize here. Definitely worth checking out, though. He seems

pretty sympathetic to the “petty bourgeois,” libertarian-capitalist segment of the free software community.

Anyway, at some point, as the saying goes, quantity gets transformed into quality:

So, what you have to be talking about instead is some way of actually spreading from this small social group here, to larger social groups. Now these are gonna be — if you’re starting to talk about spreading to other groups, and they’re probably still gonna be in these circles of unemployed, self-employed still, although not in the programming sector. You’re talking about people who don’t necessarily know how to program, or have any interest in programs, programming, or want to become programmers. If there’s going to be a world based on free software principles, it will not be a world which is entirely composed of programmers. Most people find it incredibly tedious and boring and don’t even want to understand it. So you have to talk about spreading in stages from this; you have to talk about defining, how is it possible to get to other groups? Eventually, you hope it will have spread far enough that the ideas and practices become commonplace for people. And then, the person stuck in the factory making washing machines can say, “Well, yes, but in the factory I do this, but when I’m outside here I want to listen to some music, I do something quite different. And if I need a program for my computer and it doesn’t work, I ask somebody and they tell me. Why is it in here that I do everything by orders?” And once large chunks of people’s lives have become the other way, then there’s the possibility that they might start to think, “How do I organize this other stuff that

the only way to accomplish the real revolution is by changing the way we do the most ordinary things and organize our daily lives. The political revolution, when it comes, will be only a final cracking of the shell.

Not that there's no room for political action in the meantime. But the main arena for political action is simply, as Seaman says, the law. Seaman's first priority is replacing the present neoliberal framework of IP law with a much more free software-friendly system. But the same principle can be applied more generally: push back, as much as possible, the framework of state laws that impedes our process of building counter-institutions.

I'm doing differently?" But it's not going to come all at once. So...

The application of this "free software" spirit to manufacturing, as Seamans says, has been quite common in the early, libertarian stages of modern revolutions. For example:

Now I would claim that it has been done before, that people have produced material goods on free software principles before, and quite regularly, but for very short periods of time. Basically, during every left-wing revolution this century, there has been a period where it's happened. It doesn't happen for very long: if the revolution loses, it gets squashed; if the revolution wins, it's being squashed; but it happens for a short period. There is an enormous amount of creativity, of wanting to do things, that is in people. So this is something that does not seem to get documented. I thought this would be so easy to find out about, getting lots of examples of from books. I found it very hard to get many examples of this from books, but I'm quite certain it's something that happened. I can tell from my own experience where I have seen, for example, people working in a Peugeot car factory that was under worker's control and were getting poor quality input parts coming from a French factory; communicated with the French factory and telling them, "We're running this now. Please, we're having major problems because you're sending us bad parts". And they got good parts, fixed it. Those same people then found that Peugeot didn't want anything to do, after a few months with the factory, with worker's control. They went around the area and said, "What is most in demand here?" And this

was a city on the edge of a farming area. And they got the reply that there was a real shortage of fridges, so they converted — especially for people who were in the local farms that found fridges too expensive to buy — there was a need for fridges. They converted the car plant to fridge production. God knows how they did that. I knew people who worked in there, but I don't know anything about how the thing actually worked. All I can say is, during that time, people were working on what I think of as free software principles: they were cooperating with one another to do things, asking one another what they wanted. I have no idea what the techniques are involved. I know a little bit about software, I know absolutely nothing about fridge manufacture. The fridges were a complete coincidence; it really was fridges, not washing machines; but it was very, very close.

I couldn't tell them how to do that; people find their own solutions; people in that situation find their solutions. Yep...

So, another example which I have seen more of — I should say this, to give this some context, this was in Portugal in 1975 — people building their own houses, people living in slum areas, with cooperation from architects, getting together, and on a really large scale, actually building housing estates. Very good housing estates, housing estates that I have found — I have relations who are living in one of these houses still. It's very solid; they built it very well. But it was built by local people; it was built with the design that was done together with architects. So there were people who came and gave their skills as well. But it wasn't somebody coming in from outside and saying, "I'm doing

a bit of slum clearance here. I'm gonna give you new houses." People actually built the houses themselves.

So, I mean these things can be done; but, I've said, that has been happening in the middle of revolution.

Murray Bookchin wrote an excellent multi-volume history of the phenomenon, *The Third Revolution*. In the period between the disintegration of the old central state, and the rise of either a counter-revolutionary regime or a new centralized "People's State," there's commonly an interval with quite a bit of genuine bottom-up organization: genuine power exercised by local soviets, workers' councils, neighborhood committees, and the like. In Soviet Russia, the one thing Lenin and the Whites had in common is that they didn't much care for that sort of thing (although Lenin thought the *name* was worth keeping, under new management). In republican Spain, the Madrid Communists and the Falangists competed fiercely to see who could suppress such forms of self-organization more quickly in the areas under their respective control. Interestingly, one of the first forms taken by the anti-Soviet revolts in East Germany, Hungary and Czechoslovakia was workers' councils in the factories; the victims of the Soviet bloodbath in Budapest, so beloved of the American Right, were the sort who would have been called up before the HUAC in this country.

And of course, as Seaman points out, all of those revolutions were eventually suppressed, along with the factory committees, either by counter-revolution or the triumph of the "workers' party." Whether its Kronstadt or the Santiago stadium, it's pretty much the same.

So the main lesson I'm drawing from this is that the way to get there from here is "building the structure of the new society within the shell of the old." The revolution, in the sense of a dramatic collapse of the old political order and the substitution of a new way of doing things, can only succeed if it's an afterthought to a *real* revolution built up over the previous generation or two. And