

Summer 2006

Da Torvalds Code: Opening Regional Minds to Open Source

Mateusz Perkowski

Let us know how access to this document benefits you.

Follow this and additional works at: <http://pdxscholar.library.pdx.edu/metropolitanstudies>



Part of the [Urban Studies and Planning Commons](#)

Citation Details

Perkowski, Mateusz, "Da Torvalds code" (2006 Metroscape, Institute for Portland Metropolitan Studies, Portland State University)

This Article is brought to you for free and open access. It has been accepted for inclusion in Institute of Portland Metropolitan Studies Publications by an authorized administrator of PDXScholar. For more information, please contact pdxscholar@pdx.edu.

DA TORVALDS CODE

*With the inventor of Linux here,
the Open Source community takes off*

By Mateusz Perkowski

From the ancient flint sickle to the supercomputer, tools have satisfied human needs. Even so, people have an inherent fascination with problem-solving that surpasses mere necessity. Curiosity and competition force new methods and implements to continually be created, improved upon, or supplanted by better versions – and, occasionally, the process of technological development itself is revolutionized. Prehistoric humans prompted one such breakthrough by initiating primitive speech, which in time was followed by written words and numbers. This drive to codify and transmit information also resulted in the most radical invention of our time. Instead of relying on guttural sounds or crude symbols, however, computers communicate in the language of mathematics.

In the early years, 20th-century innovators used vacuum tubes and punch card systems to operate physically enormous but retrospectively weak computers. By the 1970s, incredibly compact integrated circuits replaced these contraptions, allowing computers to be “educated” with contemporary programming languages. At the very core, though, the process remains the same. The machine translates commands into a form of data it understands on a fundamental level: the binary sequence, a pattern of 1 and 0 values that compels the computer to execute various functions.

Although built-in hardware contains the computer’s entire potential, the machine’s utility is limited unless a programmer teaches it to think by typing in “source code” – a series of directives. It is similar to parents teaching their baby to eat, walk and talk, except the computer’s Central Processing Unit is much more obedient. Both the child and the machine are predisposed to learn, but need guidance.

As lightning-quick modern computers replaced the slow behemoths of yesteryear, the pro-

gramming possibilities grew exponentially. Powerful machines and cutting-edge software, combined with a worldwide network of computer users, have resulted in unprecedented access to information. For instance, aspiring musicians can now produce albums on a tiny budget, and thanks to the Internet, broadcast their creation for the entire world to critique.

Such drastic transformation inevitably led to upheaval, as evidenced by the lawsuits brought against teenagers for illegally downloading music. For the record industry, the problem is that songs can no longer be limited to a physical form, such as a compact disc or cassette. They now exist in an ethereal realm, where no obstacles to duplication and distribution exist.

The phenomenon is new to the economic system: merchandise that can instantly self-replicate. In the world of software, this has been a predicament ever since personal computers became widely available. If a company spends millions of dollars creating a product, but each customer who buys it can give it away to an endless number of family, friends, and co-workers without any cost to himself, how does the manufacturer make any money? To mitigate the danger of this happening, software companies encode duplication limits into their programs and require license agreements. However, some people argue the mainstream system of selling information isn’t only unfair, but also inefficient.

A New Direction for Information Technology

Source code controls everything from the computer’s basic operating system to auxiliary functions, like word processing programs. Many professionals in the information technology field believe treating copies of data as commodi-

ties is inherently artificial, and the structure of software development and commerce must change. This “open source” movement doesn’t necessarily require software to be free. Rather, it calls for the underlying code to be accessible to everyone. By default, this may cause a program or operating system to cost nothing to reproduce, but that isn’t the point. The open source movement’s ultimate philosophy parallels the First Amendment of the U.S. Constitution: the exchange of ideas should not be repressed.

“If I make tractors for a living, my profit isn’t from super-secret tractor sauce. It’s because it takes a lot of metal, and a lot of people banging on it to make a tractor,” says Bart Massey, a Portland State University professor specializing in open source technology. “That isn’t true here. Once you build one tractor, you’ve got all the tractors you could ever want, infinitely. Software is treated in the law like property, but software doesn’t act very much like property.”

Massey is currently participating in a joint project between PSU and Oregon State University to promote open source research, development, and education. (The program is funded by a \$350,000 grant from Google.) From the perspective of someone involved in the movement since its infancy, Massey thinks open source will continue to succeed because it resonates strongly with the mindset of many programmers. Like fledgling novelists, novice programmers must acquire an understanding of the language. Sharing one’s creation, whether it’s a short story or rudimentary computer program, is necessary for an artist to gain feedback as well as prestige. As young programmers polish their abilities and get jobs, many continue to seek outside criticism and large-scale collaboration – the very basis of open source development.

“People have been writing and giving away software for a long time, before there were any fancy names or licenses for it,” says Massey. “The hobbyists have been doing this before there was any money in it just as a way to improve their skills, have fun with their friends, and to learn stuff. It’s quite an ego trip to put together some really great piece of code and show it to all your friends and have them be impressed by it.”

Open Source Growth in Oregon

By definition, the open source community of programmers is a nebulous society spread across the globe. Propelled by a blend of altruism and rivalry, for years they have generated computer programs outside the mainstream corporate model by communicating online. But in recent

years, open source activity has begun to coagulate in certain cities and regions, partially due to a surge in commercial investment in the technology.

Portland – along with cities in Texas, Germany, the Netherlands, Japan, and other areas – has become a prominent hub. One plausible explanation is that open source is an outgrowth of the region’s 60-year history of hi-tech innovation. Ever since Tektronix came out with the first triggered oscilloscope in 1946, Portland has proven to be a fertile ground for the technology industry. As world-wide interest in open software mounts, it’s only natural that companies with a foothold in the metroscape become stakeholders in the movement. Also, because physical logistics are practically nonexistent for open source development, the issue of livability can be a higher priority for employers who hope to be magnets for imaginative workers.

“Oregon attracts creative people who are capable of thinking in a new paradigm more easily,” says Dan Frye, vice-president of the IBM Linux Technology Center in Beaverton. The center is composed of dozens of offices on five different continents, but the local branch is home to IBM’s leading developers of Linux – a revolutionary open source operating system.

The company didn’t come to Portland specifically because of the region’s reputation as a forerunner in the field, however. “It was happenstance. IBM acquired Sequent [Computer Systems in 1999] which had world-class Unix engineers,” says Frye. “We turned them on to Linux, and as it turned out, Intel also had their Linux [base] here.” In time, it became apparent that several major technology companies in Portland understood the possibilities an open source operating system could have for hardware. In other words, they realized it would make their computers cheaper to operate.

While the absence of license fees obviously plays a part in Linux’s appeal to vendors, it is hardly the only benefit of the operating system. By liberating the software-writing process, Linux is able to tap a deep well of expertise among programmers. The creation of open source code is a community effort, but it’s also survival of the fittest. Hundreds of programmers from various companies are able to manipulate Linux, but only the most purposeful and well-designed



Portland State University's business accelerator.

changes are incorporated into the final product. Such methodology would cause complete chaos if people were building a house or car. With software,



Stuart Cohen, Chief Executive Officer of OSDL.

on the other hand, it only speeds up progress. There is no concern about keeping the information proprietary, so possible improvements can be released and widely evaluated as soon as they become available.

Another paradox of open source development is that fierce adversaries can

cooperate productively. Programmers from competing companies lack an incentive to undermine each other, simply because any malicious code would immediately be cast aside. "If IBM programmers write code that causes Linux to work slowly on Hewlett-Packard machines, the community would reject it. HP is a major competitor, but we work together on the development of Linux, and then we compete in the marketplace based on service and the best hardware," says Frye.

Through correspondence on Internet mailing lists, programmers contend with each other to earn personal recognition, and rank is determined by accomplishment instead of politics, seniority, or corporate affiliation. "Your credibility is wrapped up in it. If you write good code and give good comments, you move up in the community," he says. "If you have bad code and bad quotes, you will get burned. People will learn to ignore you."

Although hardware vendors like IBM, Intel, NEC, and Hewlett Packard, among others, saw the opportunity to profit from this unusual dynamic, the open source community was initially wary of corporate interests becoming involved in the development of Linux. Despite the enticements – such as the opportunity to test the operating system on more powerful computers – the fear within the movement was that big companies would usurp Linux and restrict the openness of the development process. To bridge the divide between corporations and Linux programmers, an independent Beaverton-based non-profit organization was formed in 2000.

"Linux developers in the open source community around the world were primarily doing this thing

out of passion. They were doing these things off hours, because it wasn't their day job. At the time, the developers didn't necessarily trust the vendors, and the vendors didn't think the developers would give them a chance. So, they formed this non-profit organization, the Open Source Development Lab (OSDL)," says Stuart Cohen, Chief Executive Officer of OSDL. "The idea was that the development community could come run their projects in a non-profit vendor-neutral environment, to see how their code was running on bigger, better machines."

Cohen says that in the ensuing years, OSDL grew to over 80 member companies that hired many of the freelance open source programmers. Linux's performance on the group's servers – potent central machines that link computer networks with databases or software programs – confirmed the operating system was perfectly suited for large ventures. The question then became how to expand open software into other media, such as cell phones, ATMs, embedded devices, and, of course, desktops and laptops.

Linux on a Global Scale: David and Goliath?

According to International Data Corporation statistics provided by IBM, the growth rate of the entire Linux market is about 26% per year. By 2008, it is anticipated to top \$35 billion. That same year, Linux is expected to claim 28% of server operating system shipments, which is eight points higher than its current share. The percentage of personal computers running Linux is still minuscule, but experts like PSU's Bart Massey believe the operating system will eventually gain traction in the consumer market as well. Generally, the population gets more computer-savvy with each passing year. As regular citizens follow in the footsteps of their geek predecessors, it's only a matter of time before they stumble upon Linux and other open source software.

"In the backroom internet server world, it's taking over. The more technically knowledgeable people are, the more likely they are to be interested in alternatives," says Massey. "One of the confusing things about open source is that there are a lot of choices. For beginners, having the choices made for you is very nice. As you get into it, you begin to appreciate flexibility."

The day when the Windows logo becomes a rare sight on computer screens isn't likely to dawn anytime soon. Even so, the climbing status of Linux among server users shows Microsoft's domination is not impenetrable. In response to this challenge from the open source movement, the corporation

has created a website, www.getthefacts.com, aimed to debunk praise for Linux. The tone isn't pushy or desperate, but the fact Microsoft went to this length indicates their level of concern. After all, grass is always greener on the other side, and customers will likely be tempted to peak over the fence. If they do, Microsoft needs to ensure they hear more than the open source side of the story.

"They are a threat, but everybody is free to compete," says Bill Allen, a partner account manager at Microsoft. "It keeps everybody honest." Allen works with subsidiary companies that help Microsoft develop and distribute software – he is not a company spokesman, so every opinion is entirely his own. From what he's seen around the office, however, it's clear the company considers Linux a force to be reckoned with.

"Microsoft management has been really open to meeting with open source customers. When the execs come to town, the first thing they ask for is, 'Take us to your latest open source customer,'" says Allen. "We want to sit down, discuss it with them, and find out where their success is. It's not a sales thing; it's just to figure out what's going on. Microsoft is always trying to figure out where they should invest."

Although the earth shifted beneath Microsoft's feet in the server marketplace, the company knows it still holds the overall higher ground. Microsoft office tools have incomparably greater momentum than even the most popular open source products, so corporations and individuals that rely on Explorer, Word, Outlook, Excel, and PowerPoint may be reluctant to part with these programs in favor of an unknown system. "Large accounts are using not just our infrastructure, but our business applications. We're growing faster in that than we are in anything else," says Allen.

He adds that clients shouldn't be swayed by the freedom from license fees because Linux and other open source programs cost more to maintain in the long run. To achieve the same level of reliable support that Microsoft provides, companies would need to hire additional employees to keep the system operating smoothly. Personal computer users, on the other hand, would be left to roam the inscrutable sphere of online mailing lists entirely by themselves. "When you use open source, you have to use pretty much whole teams of developers," Allen says.

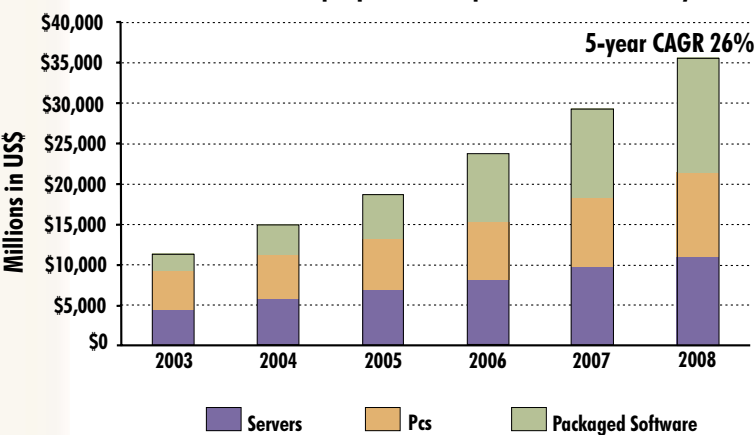
Open Source Business: Beyond Operating Systems

On the surface, Allen's observation is correct. The ability to manipulate source code and fix bugs oneself may enthrall an individual like Massey, but it seems daunting to the layperson. Scouring through thousands of lines of code to determine why a computer crashes every time a certain program is launched isn't something an ordinary person is willing to do. For some companies, however, the disparity in expertise between programmers and users of open source software wasn't perceived as an insurmountable gap. They saw it as an unfulfilled market.

Firms such as Red Hat, founded in 1993, have built their business models around selling services and support for computer programs created by other companies. The prospect of relying on an undefined community of programmers for support makes many business owners uncomfortable. Without in-house experts, they'd prefer to depend on an established entity – which is where companies like Red Hat step in. "They bundle up a bunch of free stuff, most of which they didn't write, and say, 'If you buy it from us in this shiny box with the pretty packaging, you'll also get some support, and some upgrades as they become available,'" says Bart Massey. "You get something that makes new customers or commercial customers feel more comfortable about using open source. They want somebody who they can identify as the person responsible."

Software companies also derive gain from their arrangement with Red Hat. People may subscribe for the service because of Linux, but then discover other open source tools available through the company. This up-and-coming market has compelled firms like Tripwire, a Portland-based computer security company, to release the source code to their software despite the risks. Since Tripwire's security program was first developed in the early 1990s, it had operated under licenses that allowed the source code to be reproduced for academic purposes but not sold

Linux revenues are projected to top \$40 billion in two years



Source: OSDL

commercially. As part of its partnership with Red Hat and several other companies, Tripwire officially opened its code in 2000.

According to Information Security, an enterprise security magazine, the availability of Tripwire's source code has spawned several popular clones. The companies producing these programs don't just duplicate Tripwire and sell it, but rewrite it to fit different preferences, such as removing seldom-used functions to increase efficiency. Even though these firms are basically using Tripwire's own code to work against it, it's unlikely they will achieve the company's status. Ultimately, Tripwire opened its product to an enormous market at the cost of creating a few contenders that don't truly threaten its power. "Ordinarily, an influx of open-source competition would be seen as a threat to a commercial vendor," writes Pete Loshin in Information Security. "However, Tripwire seems relatively unconcerned about rivals nipping at its heels."

Smaller software companies also face the possibility of competitors appropriating their open source code, but without Tripwire's longevity, the eventuality can be disastrous. Their best recourse is simply to advance their product at a pace that is unfeasible for another company to keep up with – the strategy adopted by Compiere, a software firm based in the Portland State Business Accelerator. The company's business model is atypical because no emphasis is placed on selling software. They focus completely on development and maintenance, and use partner companies as their sales arm. In effect, this symbiosis detracts companies from cloning Compiere's software. Rather than subsume the source code, firms that want to make money from selling services for Compiere can simply enlist as partners. While competitors would need to worry about hiring engineers to update and build upon the software, partner companies leave all that to Compiere.

"It doesn't make business sense to take [Compiere's source code] and run with it. If you take it and run with it, you'd have to do something that you'd get otherwise for free," says Jorg Janke, the company's founder. "There's only 24 hours in the day, and you need to concentrate on different things. It simply makes more business sense to concentrate on the value-added, rather than the basic."

Compiere produces both Enterprise Resource Planning (ERP) software for general business management and Customer Relations Manager (CRM) software for sales-assisted processes in websites or call centers. When customers sign up for an ERP or CRM bundle from one of 60 partner companies worldwide, their payment includes support and ser-



Jorg Janke, Compiere's founder.

vices from Compiere. Although Compiere provides the program's basic framework, its partners are free to develop software add-ons to deal with their clients' specific needs – depending on the license, even proprietary add-ons can be created for open source software. This poses no problem to Janke, since it expands the software's function without investment from Compiere. "We are an enabling technology, so the more usable it is, the better it is for us," he says.

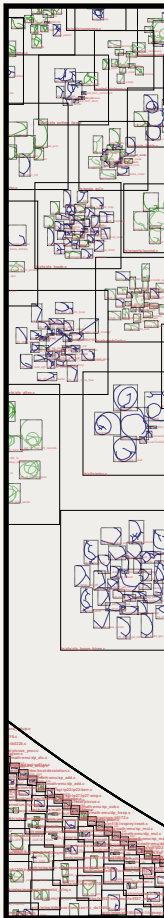
The main reason Janke decided to start an open source company was to widen his products' availability; if people could have top-notch software for free at the click of a button, Compiere was bound to catch on. So far the theory has proven to be correct –there have been over a million downloads of the software, making Compiere "the premier open source business application," Janke says. "The main motive for us was to get very broad exposure. If you just build a better mouse trap, it doesn't mean you're going to have success, unfortunately. Open source was our way into the market. That exposure would be very difficult to get with a traditional model."

The Future of Open Source

To enhance the marketability of open source products and promote economic activity in the region, in early 2005 the City of Beaverton launched an incubator, the Open Technology Business Center (OTBC), with \$1.2 million in seed money. "Whether it is administrative support, PR and marketing, communications, executive coaching, product strategy, there are many different pieces



Holly Files, executive director of OTBC.



that are being brought to the table” for fledgling open source companies, says Holly Files, the organization’s executive director.

Startups that seek help from the nonprofit don’t necessarily have to create open source software, but do need to incorporate it into their business. Currently, OTBC houses several new companies that use open source software for very different applications. For example, Infinity Softworks has developed a subscription service that features tools and modules to help students learn math, while OregonStart-

ups.com hopes to be an online portal for new technology companies. By helping such small firms develop some legs, Beaverton is banking on the energy of the open source movement to reinvigorate the region’s hi-tech economy.

“A company that starts in an incubator has 70% chance of being in existence four years later. For a company that starts singularly on their own in an office or garage, the odds are about 20%. Your odds of success are extremely high when you come into an incubator,” says Files. “The second thing that’s interesting is that companies that graduate successfully out of an incubator tend to stay in a five or 10 mile range of that incubator.”

At this stage, it is too early to predict the impact OTBC and the open source movement in general will have on the Silicon Forest. The local technology industry still hasn’t recovered from the economic downturn that followed the hi-tech boom of the 1990s, so a great deal of hope has been vested in open source. The optimism isn’t unfounded. Unlike the dot-com craze, open source seems to be setting firm roots in the metroscape.

Now that the open source movement has matured into a full-fledged sector of the technology industry, the society has changed. Unlike the geeks who initiated it, business professionals have a culture in which face-to-face meetings in corporate office buildings cannot be replaced by mailing lists and newsgroups. As these “suits” become more involved in the open source movement, business activity will correspondingly coalesce in certain regions. The Portland area’s history of technological innovation, combined with a pleasant environment and overall lifestyle, makes it an attractive option for companies geared toward open source development.

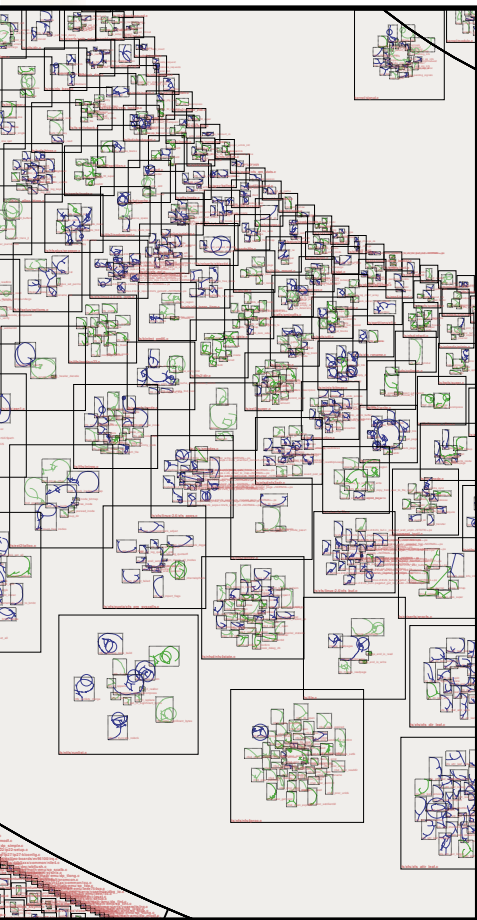
The arrival of Linus Torvalds, creator of Linux and de-facto development leader of the operating system, is also cited as a sign Portland is an emerging gravity center for the open source movement. In 2004, Torvalds moved to Portland’s suburbs after a year spent working long-distance for OSDL. Considered the high guru of open source, Torvalds embodies the spirit of the movement: he programmed the kernel for the Linux operating system as a reclusive geek living with his mother. During the grueling months he spent in front of his computer designing the system, money and fame didn’t cross his mind – Linux was written purely for fun.

Torvalds attributes the fact that his hobby turned into a multibillion-dollar industry to this laissez-faire attitude. If he had been possessive and unreceptive in the development process, he’d still probably be the only one using the operating system. Openness, on the other hand, gave Linux the capability to develop a life of its own.

“It’s like letting the universe take care of itself. By not controlling the technology, you are not limiting its uses,” writes Torvalds in *Just for Fun*, his autobiography. “This is not about trying to spread Linux. It’s about making Linux available and then letting it spread itself. And this doesn’t apply only to Linux. It applies to any project that’s open. Open source makes sense.”

Torvalds essentially shares the same philosophy as the ancient inventors of human language: communication fosters knowledge and growth. According to him, computer programs and operating systems can become greater than the individual or corporation that created them, but only if they don’t try to suppress the data out of fear or greed. In this view, software running on open source code must naturally evolve and thrive – precisely because it is better adapted to the uncontrollable nature of information. **M**

Mateusz Perkowski is a contributing editor for MetroScape and a Portland area freelance writer.



A portion of a diagram of Linux's kernel. Courtesy of OSU.