

Free, Libre, Open Source Software and Creativity in Programming

by

George Grinstead (MA Interaction Design)

Draft I.

Computer Programming has traditionally been seen as a field of science and taught under the titles Computer Science and Software Engineering. In recent years, however, the expansion of the Open Source software movement has meant that programming has become a much freer, humanistic field. This paper looks at whether programming is now more akin to a field of art, design or even politics.

History

Programming initially emerged as a science from electrical engineering's creation of computing. During the early days, software was written for specific hardware with computational efficiency as the goal. This necessity of machine dependant honing meant that programming seen a science with discrete examples of effective software – the science was in discovering the most efficient means of programming a specific machine to do something. It was not until 1978 that computer software became machine independent but to see where the Open Source movement began we need to travel a little further back.

The history of Open Source software emerges from the history of the Hacker, a term first used circa 1961¹ and later appropriated by the media to describe a Cracker or computer criminal. Initially adopted by MIT's Tech Model Railroad Club, a group of early computer enthusiasts that later founded MIT's Artificial Intelligence Laboratory. The 'correct' definition of Hacker now stands as:

...any kind of expert, especially with the connotation of having particularly detailed knowledge or of cleverly circumventing limits.

wikipedia.org (2005)

In 1969, the originally disparate Hacker communities of MIT, Stanford University and Carnegie-Mellon University were linked by ARPAnet, the experimental beginnings of the Internet. Through this connection, the Hacker communities at these three academic institutions, alongside those at XEROX PARC and Bell Laboratories are jointly responsible for most of the technical innovation that lead to the personal computer revolution.

What can be seen from this point on is that the Hacker class, as described by Wark², occupies a unique position as the only productive class that designs its own tools of production. This ability has meant that throughout the history of the commodification of Hacker output, Hackers have always been able to steer development somewhat 'their way'. As the communities mentioned above were encouraged by their Vectorialist rulers, the universities and corporations, to experiment with ARPAnet they began to develop means of serving their own interests at the same time. From 1970 on, ARPAnet's ability to support mailing lists was used not only for academic/commercial collaboration but also for socialising, the most popular list being SF-LOVERS, a list for science fiction fans. 1973-75 on ARPAnet saw the first remote collaboration on a text document – the first Jargon File, a slang dictionary of Hacker terms³ since published twice, once as *The Hacker's Dictionary* and then again as a revised and expanded edition *The New Hacker's Dictionary*. This dual development of personal and vocational interests intertwined in within the same projects can be seen throughout Hacker history and will be covered in much more depth later in this paper.

The next major event in Hackerdom was the development of the Unix operating system, actually occurring alongside the development of ARPAnet, from 1969 onwards at Bell Laboratories. Whereas ARPAnet was allowing the disparate Hacker class to achieve critical mass, Unix was changing the very emphasis of programming. Based on an operating system called Multics, Unix aimed to hide the complexity of an operating system from the user and programmer making the operation of the computer and further programming much easier. This is the first fundamental change in the 'science' of programming where ease of use takes over as the primary objective. With Unix, the stated goal was

to make using a computer easier not more computationally efficient, this switch was only possible because Thompson and Ritchie, the developers of Unix and the C programming language realised that computer hardware and compiler technology had become advanced enough for an operating system to be written in a higher-level, less efficient language. By writing Unix in C, the first machine-independent operating system was born and along with it the ability for Hackers to move chunks of working code from machine to machine, project to project.

With Unix fast becoming the Hacker operating system of choice due to the ability to reuse and build on code from project to project, the Hacker movement was now centred around the same operating system and the new social networking possibilities it provided in Usenet, a much more inclusive bulletin board system that soon grew larger than the more exclusive ARPAnet. It is at this point that Richard Stallman of Bell Laboratories tried to unify the community politically with the creation of the Free Software Foundation and an entirely free clone of Unix – Gnu's Not Unix (GNU).

The Free Software Foundation's ideology was basically the previously unwritten goals of all Hackers distilled into a set of rules for software distribution, at its centre, that users should be able to examine, copy, manipulate and redistribute the code used to create any piece of software. It is another common misconception that the Free in Free Software Foundation stands for monetarily free, since its conception it has been meant to refer to user freedom. The inception of the Free Software Foundation is, perhaps, the most important event in the history of the Hacker as it set in motion the transition of the Hacker ethic from a relatively insular clique to a publicly facing politic, a politic whose basic tenet is the right to manipulate or hack and build on the work of others who believe the same.

Between the founding of the Free Software Foundation in 1982 and 1991 its proponents produced many of the harder parts of a free Unix but had failed to complete HURD, their version of the Unix kernel, the central core of the operating system. At the same time attempts by AT&T and Berkley Software Design to commercialise Unix were failing and Microsoft was able to take a large percentage of the personal computer market with the technically inferior Windows operating system. As the death knell was being called for Unix, and with it, the seeming success of Hackerdom a University Student at Helsinki University single-handedly, though laziness, created a new model of software production that would forever change not only Hackerdom but also the commercial software world and beyond. In 1991, Linux was born.

¹ Raymond, E. (2001) *The Cathedral & The Bazaar*, California: O'Reilly

² Wark, McKenzie (2004) *A Hacker Manifesto*, Massachusetts: Harvard University Press

³ Raymond, E. (2001) *The Cathedral & The Bazaar*, California: O'Reilly

Sections to Follow:

The Internet – Allowed the creation of disparate development communities and thus Linux.

Linux – First remotely produced software.

Netscape – First commercial adoption of Open Source model.

Wider Commercial use of Open Source Model – IBM, Apple, Redhat, etc.

Copyleft – Using Copyright to protect freedom not personal gain.

Creative Commons – Multiple Copyleft licenses.

Gift Economy/Bounties – Ubuntu, Mozilla, Firefox advert.

FLOSS and Design – Firefox, Camino, etc.

Other Fields – Science, writing, music, video (archive.org and BBC).

Contrasting Against the Cultural Industries – Gift economics vs. commodification.

Copyleft as Politics – Infectious not revolutionary, ground up, producing class orientated.

Conclusions – Creativity in software production, programmers as designers, FLOSS politic potential.

Threats to Conclusions – Hardware/Internet reliance.