Chapter 4

Open Source and Free Software Licenses for Embedded Systems

Renê de Souza Pinto
Zededa GmbH, Germany

ABSTRACT

Several open source operating systems are being used in commercial embedded applications, such as smartphones, set-top boxes, routers, video game consoles, and many other consumer electronics. One of the best examples is the Linux kernel, which is present in millions of different embedded devices. That said, the Linux kernel is licensed under GNU General Public License version 2 (GPLv2), which enforces any derivative work to be licensed under the same terms as the original license. Embedded device manufacturers must be aware of such clauses and understand how to handle the distribution of their systems, including free software and open source operating systems, such as Linux kernel. This chapter covers relevant aspects of different open source licenses regarding operating systems and some common issues faced by developers of commercial applications. The foundation knowledge is presented to guide readers to choose an open source operating system according to its license for embedded commercial applications.

INTRODUCTION

The Operating System (OS) plays a central role in a computational system: it provides an abstraction of the hardware for running applications and manages all machine resources. A few decades back, Operating Systems were evolving along with the

PC industry, while embedded systems were still composed of very specialized applications, usually running on microcontrollers with much less computational power than general-purpose systems. However, the rapid growth of microelectronics technologies pushed embedded systems to a new level: embedded platforms emerged on the market using powerful SoCs (System-on-a-Chip) that can run general-purpose OSes with a performance comparable (or in some cases, even better) with PCs. Additionally, the embedded hardware became less expensive and the FOSS and Open Source Hardware community brought to life many open embedded platforms through successful projects, such as the Raspberry PI. Currently, a huge number of embedded platforms running FOSS artifacts (such as the Linux kernel) are inexpensive and available to the general public. This evolution was a key factor to the development of new technologies, such as the development of IoT devices and has changed the embedded industry as well. Manufacturers started to use FOSS in their products (routers, smartphones, TVs, among many others) and several licensing issues arose throughout the years. In 2008, Free Software Foundation (FSF) initiated a lawsuit against Cisco claiming that GNU's GPL (General Public License) and LGPL (Lesser General Public License) were violated in many Linksys (acquired by Cisco) products (Lee, 2008). GNU's tools *binutils*, *coreutils*, *glibc*, among others, were distributed only in binary form along with several Linksys firmwares without meeting all requirements fixed by these licenses. In 2009 the parties announced a joint agreement that included a non-disclosed monetary contribution from Cisco to FSF (Smith, 2009). In 2006 the German programmer Harald Welte prosecuted D-Link (iFross, 2006) claiming that software of his authorship that were part of Linux kernel was distributed along with the firmware of a storage device (Wireless G Network Media Storage DSM G600) without meeting all GPL requirements. The violations were found after a reverse engineering process made by the author, which bought a device. The court ordered the reimbursement of all costs expended by the author with the legal process, purchasing of the device, and reverse engineering work, besides enforcing D-Link to meet all license requirements. The same programmer was also the plaintiff in other cases brought to court in different countries. In 2013, Welte prosecuted the company FANTEC for using his software kernel components (netfilter/iptables) in one of their firmwares (FANTEC 3DFHDL Media Player device) without the release of complete corresponding source code (Welte, 2013), in this case the source code was released but it did not fully match the version present in the firmware. The court decided a penalty fee to FANTEC plus expenses for the lawyers. Additionally, it was requested that the company disclosed the exact information about the media player firmware. These examples show how not following all FOSS license requirements can bring not only image but also financial damages to the companies. Although the source code is open and widely available, there are important differences on license terms across different FOSS licenses. One key

Related Content

Software Defect Prediction Using Genetic Programming and Neural Networks
Mohammed Akourand Wasen Yahya Melhem (2017). *International Journal of Open Source Software and Processes (pp. 32-51).*
www.irma-international.org/article/software-defect-prediction-using-genetic-programming-and-neural-networks/203647

Open Source Software Systems: Understanding Bug Prediction and Software Developer Roles
R. B. Lenin, S. Ramaswamy, Liguo Yuand R. B. Govindan (2010). *International Journal of Open Source Software and Processes (pp. 28-47).*
www.irma-international.org/article/open-source-software-systems/53876

Empirical Evaluation of Bug Proneness Index Algorithm
Nayeem Ahmad Bhatand Sheikh Umar Farooq (2020). *International Journal of Open Source Software and Processes (pp. 20-37).*
www.irma-international.org/article/empirical-evaluation-of-bug-proneness-index-algorithm/264483

Open Source Assessment Methodologies
Barbara Russo, Marco Scotto, Alberto Sillittiand Giancarlo Succi (2010). *Agile Technologies in Open Source Development (pp. 302-310).*
www.irma-international.org/chapter/open-source-assessment-methodologies/36509

Enhancing the Software Clone Detection in BigCloneBench: A Neural Network Approach
Amandeep Kaurand Munish Saini (2021). *International Journal of Open Source Software and Processes (pp. 17-31).*
www.irma-international.org/article/enhancing-the-software-clone-detection-in-bigclonebench/286650