Linux Kernel Development: Case study of what goes into building an open source product

Preeti U Murthy
Software Engineer
Linux Technology Center
System and Technology Group,IBM

Challenges

- There are hundreds of developers working on Linux from all across the world. How do they co-ordinate among themselves?
- Are their any conflicts that arise among the developers?
- Who stops bad code from getting into the kernel? Is there a manager or a leader?
- Do they work for free? If so how are they motivated to do so?

Developers, Maintainers and Linus

- Developers are the programmers who contribute to Linux kernel development. They work like in any other software development model. Write code and publish it.
- Maintainers are the gatekeepers of the Linux kernel. They review the code published and decide if it needs to go into Linux.
- Linus Torvalds then picks the code reviewed and approved by the maintainers to be merged into Linux.

Mode of Communication

- Mailing List, IRCs, GIT

  - Mailing List: lkml.org is the discussion forum for kernel code changes, also called patches. Developers post their patches here where the maintainers review.

  - IRC – Internet Relay chat is a messaging client where developers discuss the correctness of patches with the maintainers if they see the need for a one on one discussion.

  - GIT – Version Control tool which enables the maintainers to keep track of the changes made by various developers and the overall change in code made from one kernel version to the next.

GIT
The backbone of kernel development

- GIT allows one to maintain a code repository and manage it. This code repository is also called the GIT tree.
- GIT branches carry code which belong to different versions or topics in the tree. This is left to the owner of the repository.
- GIT branches can be pushed or pulled. Meaning, GIT branches can be merged into each other within a tree and between trees.

How GIT branches and trees are used in kernel development

- Each maintainer is responsible for a specific sub-system of Linux, like file system or memory or storage. This will be his area of specialization. He can have several GIT branches containing code in these areas.
- Developers contributing code to any of these areas will get their code into the appropriate maintainer's GIT branch after review.
- The maintainer's GIT branches containing the code are pulled by Linus finally into Linux and merged with the other maintainer's GIT branches.

Reviewing Code

- This is a very important phase of kernel development.
- Considering there are many contributors to any sub-system of the kernel the maintainers alone cannot review all of the code.
- Every developer in the community has equal privileges to voice his opinion about a code and review it. Its in fact the duty of a good developer to help the maintainers review code and ease their work.
- However a maintainer's verdict is final. If he thinks the code cannot get in, it just cannot. But his decision is sound. These maintainers are among the top developers of the world and they give value to the opinion of the developers.
- When any developer's code gets into Linux, his code is acknowledged with a a “Signed-off-by: Developer” in the changelog of Linux. So every person's contribution is solely his'.

Resolving Conflicts

- There could be conflicts between the developers about the soundness of code but remember that its a technical debate and the one with the right reasoning will win and none can stop him.
- As long as one has data points, results and a good technical explanation by his side, he can always have his way and this has been proved time and again in the kernel development.

Who are these contributors?

- The maintainers and developers can be independent programmers or they could be employed by organizations.
- The top contributors today to the kernel are from RedHat, Intel, Linaro, Texas Inst., IBM, Google, Samsung, Nvidia and many more.
- However around 12% of contributors are independent programmers.
- What motivates these programmers to stay in Linux kernel development as part of their career for many years? Some as long as 10 to 15 years!

  - The discipline maintained in the kernel development by adhering to the best coding practices in software development is astounding.

  - The Linux kernel is an embodiment of some of the best operating system concepts and continues to have scope to include support for the cutting edge technologies of today.

  These are sufficient reasons to attract programmers who code for the joy of programming and for the thrill of being in midst of happening and best technology, to continue contributing to the kernel.

Result of open sourcing Linux?

- Growth of Linux:

Year	Lines of Code (approx)
1991	10,000
1994	176,000
1995	300,000
2001	1,800,000
2003	5,000,000
..	..
2013	15,000,000

Result of open sourcing Linux?

- There are around 30 versions of Linux released so far.
- More than 7800 developers from 800 different companies have contributed to the Linux kernel so far.
- When development is being sustained at such a massive scale it surely means that its going in the right direction.
- Linux is being used from web servers to personal devices to automobiles. Its users include students and enterprises. So broad is its influence.

Other Popular Open Source Projects

- OpenStack
- OpenDaylight
- KVM
- Apache
- Hadoop
- Ovirt
- Node.js

- The list can go on to be a long one! They span from storage to networking to web development and cloud. Clearly open source projects are gaining momentum like never before.
- Note that the listed ones are popular not merely because they are open source but also because they are influencing the growth of technology today.
- They are competing against their closed source counterparts.

So go ahead and be a part of the open source revolution!

All you need to do is:

- Choose an open source project that interests you.
- Download the source code and take a tour through it.
- Understand the working culture of the community.
- Get involved in mailing lists and discussion forums to find out the pressing problems that the community is trying to solve.
- Choose a problem that interests you.
- Take help of the community when you find the going hard. Seek clarifications from them.
- Understand the procedure they follow for contributing code.
- Most importantly keep your efforts sustained. It might be hard in the beginning but like any other difficult task it gets easy as you as you spend more time on it.

References

- LinuxFoundation-releases-annual-linux-development-report
- “The Success of Open Source” by Steven Weber
- Wikipedia

QUESTIONS ?

Legal Statement

- Copyright International Business Machines Corporation 2013.
- Permission to redistribute in accordance with Linux Foundation Collaboration Summit submission guidelines is granted; all other rights reserved.
- This work represents the view of the authors and does not necessarily represent the view of IBM
- IBM, IBM logo, ibm.com are trademarks of International Business Machines Corporation in the United States, other countries, or both.
- Intel is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.
- Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.
- Other company, product, and service names may be trademarks or service marks of others.
- References in this publication to IBM products or services do not imply that IBM intends to make them available in all countries in which IBM operates.
- INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you. This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.