Linux Development Process

The kernel development cycle has evolved so beautifully overtime that it has set an example in the open source world. Having contributed to the kernel I actually enjoyed learning about the whole development cycle. Terms like mainline kernel, rc, stable release, long-term support confused me a lot initially but with time I understood at least the basic work-flow.

Keeping in the mind the volume of code that sits inside the kernel, it is very difficult for a single person to inspect each and every part of the project perfectly. Hats off to Linus and people like Greg. To make the process easier (it looks easy :P), the kernel is broken down into subsystems with each subsystem having its own main developer or as generally said top level maintainer. These maintainers decide which patch goes to the mainline kernel. Along with these top level maintainers there are many file system and driver maintainers who review the patches before the subsystem maintainers accept it. These maintainers can also send a pull request to the subsystem maintainer if the subsystem is again divided into sub-sub-systems. And that’s the reason the whole kernel is built on chain of trust. Finally all the patches collected by the subsystem maintainers goes to Linus in a pull request. So, to get your patches into the kernel send it directly to the maintainers and this is when the get\_maintainers.pl script comes handy.

Subsystem maintainers collects patches ahead of time and send them to Linus when the merge window opens. To explain more technically, let’s assume that the development cycle for version 5.1 has ended and it has released. As soon as 5.1 is released a new development cycle for 5.2 version starts and the merge window opens. Merge window is the starting point of the development cycle where the code that is considered to be sufficiently stable gets merged into the mainline kernel by Linus. All the new features for the next kernel version goes in this merge window. This is usually a two week process and the maintainers send PR to Linus and Linus merges them to the mainline kernel. But this is not how a new version gets released.

At the end of two weeks Linus declares that the merge window has closed and now the process of stabilizing the mainline kernel begins. At this point some features are still untested and unstable and some bug fixes have to be made to prepare the kernel for the next stable release. Closing the merge window means that now is the time to start releasing the -rc kernels. Over the next 8–9 weeks, the developers work continuously on the -rc versions. Every week a new -rc is released with the name 5.2.rc1, 5.2.rc2.. and so on. These -rc versions are improvements or bug fixes to the mainline kernel that got built during the merge window. During these 8–9 weeks only patches including some improvements goes to the mainline. No more features are added after merge window closes. Finally after the end of these -rc weeks the kernel is stable and ready for release as version 5.2. So, the whole development cycle is a matter of around

10–12 weeks and we get a new version in every three months.