Introduction

Abdul Rahim

July 19, 2024

Basic Info

- small incremental changes that add new features, make enhancement, fix bugs; are called **patches**
- new release in 2+ months
- new development and current release integration cycles run in parallel

Kernel release process

- lets call general updates made available to the public as launches
  - using this defination there are 2 types of launches
  - there is a **2 week merge window**, in which new features are accepted
- A major launch with **new features and fixes** is called a release
  - this is called **rc1**
- minor launches with **bug fixes only** called release candidates
  - once a launch is out in the public, it unravels bugs and security issues
  - these bugs/vulnerabilities are fixed with patches, and these paches are accumulated for a week and launched as minor launches; which is named like **rc2, rc3, rc4 ...**

- usually the process of sending and including patches goes on upto **rc7, rc8**. Subject to weather **the code is good enough**?
- After that a 3 week quiet period starts. In this time, the maintainers prepare their trees, to send a pull request to linus
- After this 3 week period, the next merge window starts, in this merge window developers send their trees to linux via **signed pull**

Kernel release process

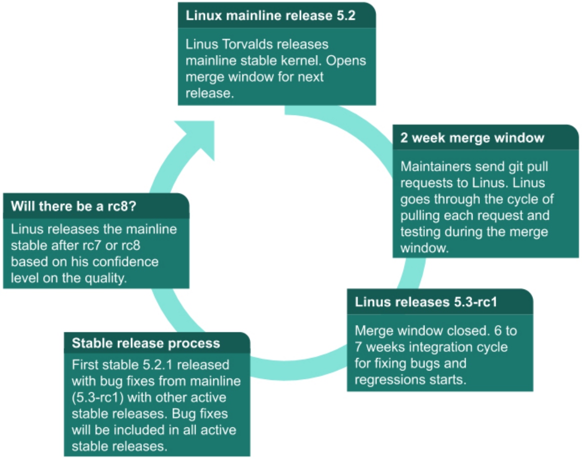
Linux mainline release 5.2
Linus Torvalds releases mainline stable kernel. Opens merge window for next release.
2 week merge window
Maintainers send git pull requests to Linus. Linus goes through the cycle of pulling each request and testing during the merge window.
Linus releases 5.3-rc1
Merge window closed. 6 to 7 weeks integration cycle for fixing bugs and regressions starts.
Stable release process
First stable 5.2.1 released with bug fixes from mainline (5.3-rc1) with other active stable releases. Bug fixes will be included in all active stable releases.
Will there be a rc8?
Linus releases the mainline stable after rc7 or rc8 based on his confidence level on the quality.

Figure 1: Release Process in the linux kernel

definations

- **release candidate/RC** releases are mainline kernel pre-releases that are used for testing new features in the mainline. These releases must be compiled from source. Kernel developers test these releases for bugs and regressions
- **stable releases** are bug fix only releases.
  - After a mainline kernel is released it moves into stable mode
  - bug fixes to stable kernel are backported *from mainline to stable* by designated stable kernel release maintainer
  - stable kernel release updates are released on average once a week(rc story)
- **long-term** releases are stable releases selected for long term maintenance
  - provide critical bug fixes to older kernel trees
- docs

kernel trees

- the kernel code is organised in several **main** and **subsystem repositories** called trees
- 1. **mainline kernel tree**: This is where linux releases mainline kernels and RC releases
  2. **stable tree**: this tree is maintained by greg kroah hartman. The tree consists of stable release branches, stable releases are based on this tree
  3. **linux-next**: this is the **integration** tree. code from large number of **subsystems trees** get pulled into this tree periodically, and then released for integration testing. This process of pulling changes from various trees catches merge conflicts between trees

- each subsysystem has its own tree and a maintainer(s), can find them *here*
- each subsystem has a mailing *list*
- development happens over mailing list:
  - contributors send patches to mailing list through emails and then they are discussed in mails
  - Once applied; a message is recieved “Thanks Applied”
  - *doc*

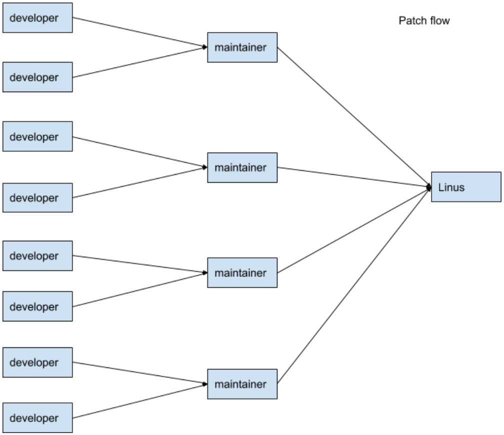

Figure 2: flow of patches