Case Study: Linux Kernel

COMP8440: FOSSD
Lecture 10

What is the Linux kernel?

- A portable operating-system kernel
  - Runs directly on the hardware and manages access
  - Includes device drivers, network stacks, filesystems, schedulers, memory management, etc.
- Who uses it?
  - Usually combined with other FOSS components to form a complete operating system
  - Widely used on supercomputers, servers and embedded devices
- Who develops it?
  - Project leader is Linus Torvalds
  - Tens of thousands of developers worldwide contribute
  - A 2009 study found that since 2005, over 5000 developers from nearly 500 different companies had contributed

How big is it?

Large project with fast pace of development
As of version 4.5, released 13 March 2016:

- 575,548 git commits since April 2005
- 13,173 git commits since version 4.4 (10 January)
  - 12080 non-merge commits, 1093 merge commits
  - 1670 contributors
- 52,722 source files, 21,153,879 lines of source
  (4,328 files, 328,446 lines in Documentation directory)
- Support for 30 different architectures
  - Not counting 32b vs 64b variants as distinct – x86, arm, powerpc, sparc, mips, parisc

Why is it successful?

Modular architecture
- Components can be included or excluded at build time
- Many components can be built as modules

Good performance and scalability
- Scales from tiny embedded devices to supercomputers
- Scales from 1 CPU to thousands

Modern features – some examples:
- Journalling filesystems
- Logical volume management and software RAID
- Extensive network filtering support

Easy to extend

Free!

First Announcement

Started by Linus Torvalds in Finland in 1991

From: torvalds@klaava.Helsinki.FI (Linus Benedict Torvalds)
Subject: What would you like to see most in minix?

Hello everybody out there using minix –

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).
...
Linus (torvalds@kruuna.helsinki.fi)

PS. Yes – it's free of any minix code, and it has a multi-threaded fs. It is NOT protable (uses 386 task switching etc), and it probably never will support anything other than AT-harddisks, as that's all I have :-(.

Early Days

- Hobby project
  - One maintainer, hosted on university servers
  - Simplistic copyright notice, prohibiting charging of any fee for copies, not even “handling” costs (changed to GPL in 1992)
  - No grand plans
- Why did it succeed?
  - It was immediately useful
  - The code was easy to hack on (around 10k lines of C)
  - It was free
  - No lawsuits!

Growing community

- Small but growing developer community
  - over 100 developers by 1993
- Code released as tarballs (.tar.gz files)
  - frequent releases, every couple of weeks
  - diff from previous version also provided
- Contributions as patches on mailing list
  - Linus integrated them manually into his tree
- Version 1.0 released March 1994 (176k lines)
- Version 2.0 released June 1996 (778k lines)
- Odd/even minor version number scheme
  - Even minor versions (2.0, 2.2 etc.) stable series
  - Odd minor versions (2.1, 2.3 etc.) development

Growing maturity

- Support for multiple architectures added – 1995
- Support for multiple CPUs (SMP) – 1996
- Growing corporate interest – 1998 onwards
  - e.g. IBM announces billion-dollar investment – 1999
- Dot-com boom – 1999–2001
- Improvements in scalability and performance – 2001 onwards
- Linux server market exceeds rest of Unix market – 2012
- Linux takes over embedded OS market – early 2010's

What SCM is used?

Git!

- Git was written by Linus Torvalds and others with the intention of using it for the kernel
- Speed and scalability requirements for git were driven by the needs of the kernel community

Git trees are hosted at git.kernel.org

Who has commit access?

In one sense, only Linus Torvalds

- Only he can commit to his repository

In another sense, every developer!

- Every developer has commit access to their own local git repository

Linus pulls commits from trusted maintainers on request

- Subsystem maintainers – networking, file systems, scheduler, memory manager, etc.
- Architecture maintainers – x86, arm, powerpc, etc.
- Some areas have a team of maintainers rather than a single individual
- Maintainers are responsible for code quality in their area

How are releases managed?

New release every 9–13 weeks

Versions numbered *x.y*

- *x* is major version number (currently 4)
- *y* is minor version number

The two weeks following the release are the *merge window*

- This is the time for maintainers to ask Linus to pull their trees
- Maintainers generally close their trees to new features when the merge window *opens*

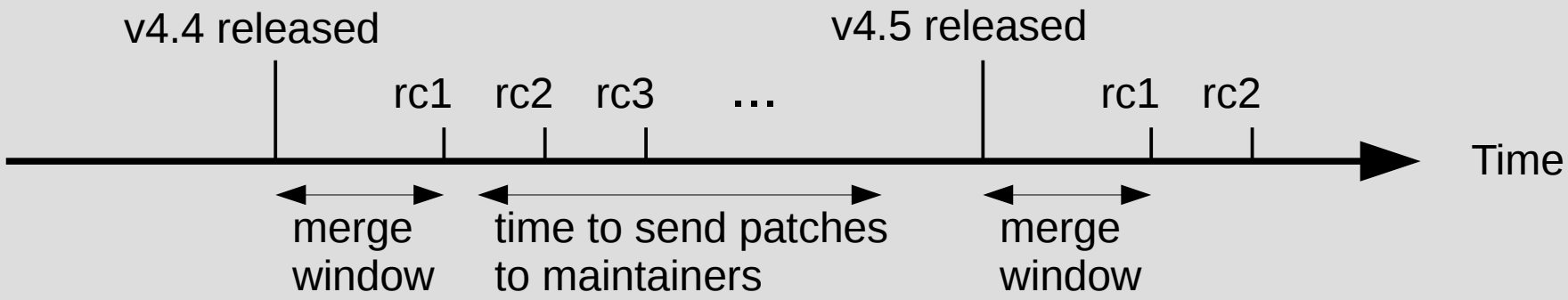
After the merge window ends, only bug fixes are accepted by Linus

- Time for stabilizing and fixing bugs

How are releases managed?

Series of release candidates, one per week

- *x.y*-rc1, *x.y*-rc2, etc.
- Later rc's only accept fixes for serious bugs and regressions



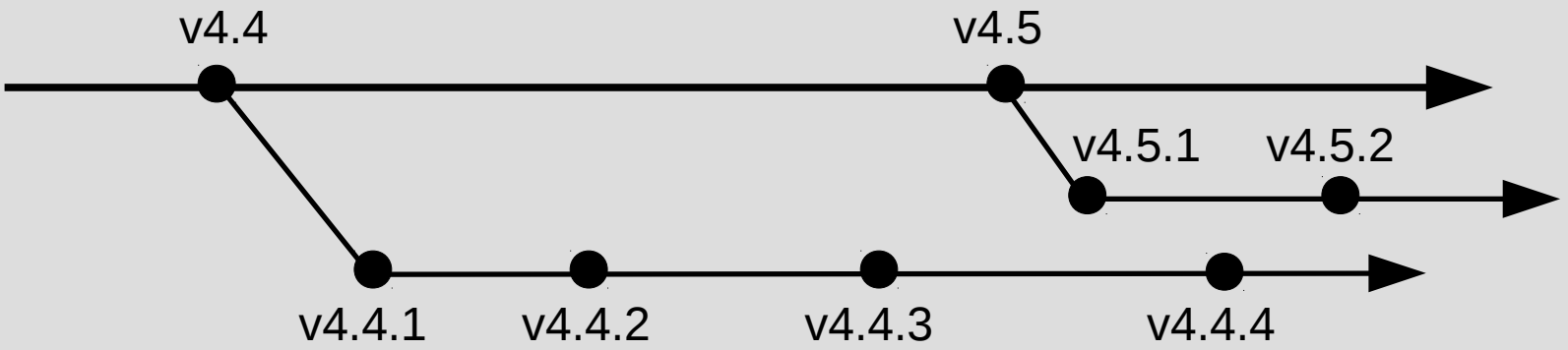
Stable vs. development trees

Each release is maintained for a period of time

- At least until the next release
- Some releases maintained for years
- Version numbers *x.y.z*, e.g. 4.5.1, 4.5.2, etc.

This series of versions is a branch in git

- Maintained by a volunteer, not Linus
- Strict rules about what patches can be accepted into stable trees



Communication methods

Many mailing lists

- Most hosted at vger.kernel.org
- linux-kernel mailing list – very high volume
- Mailing lists for subsystems
  - e.g., linux-mm, linux-ext4, kvm, linuxppc-dev
- Mailing list for architecture maintainers
  - linux-arch@vger.kernel.org

IRC channels

- #kernel on irc.sekrit.org

Kernel Summit meetings

- once per year
- attendance by invitation only, based on recent contributions and activity in discussions

Corporate involvement

Many companies want to contribute

Linux Foundation provides resources

- www.kernel.org servers – git, web, ftp servers
- Employs Linus Torvalds and some other developers
- Provides training and certification of Linux sysadmins
- Organizes conferences – Kernel Summit, Plumbers Conference, LinuxCon, Collaboration Summits, etc.
- Funded by companies including IBM, HP, Intel, NEC, Qualcomm, Samsung, Huawei, Fujitsu, Oracle and many others

Linux Foundation does not direct the technical direction of Linux development

How are decisions made?

Code and patches

- Review and discussion on subsystem mailing lists, with final decision made by maintainer

Processes and policies

- Discussions on linux-kernel mailing list
- Discussions at Kernel Summit
- Final decision with Linus Torvalds

Releases

- Time-based
- Linus decides which -rc will be the last before release
  - based on rate of bugfix patch arrival

Features

- Whatever contributors want to work on...

Project policies

Coding style

- see Documentation/CodingStyle
- scripts/checkpatch.pl
- small amount of variation between subsystems

Patch submission guidelines

- see Documentation/SubmittingPatches
- Signed-off-by required

Project documentation

Many disparate places:

- Documentation/ directory in source tree
- Man pages project
- Many and various web pages
- Many books

Project roles

Leader and BDFL – Linus Torvalds
- Effectively also release manager

Subsystem maintainers
- Generally handed along rather than voted on
- Some subsystems have a maintainer team

No formal bug master

No formal web master
- Some web pages maintained by Linux Foundation

No formal documentation master
- Informally, Michael Kerrisk maintains the man pages

Mailing lists manager – David Miller