Maintaining a Linux Kernel for 13 Years? You Must be Kidding Me. We Need at Least 30!

Ben Hutchings / Agustín Benito Bethencourt
ELCE, Prague, 24th October 2017
The speakers:

- **Ben Hutchings**
  - Kernel developer at Codethink Ltd.
  - CIP kernel maintainer (4.4)
  - Debian and stable (3.2 and 3.16) kernel maintainer

- **Agustín Benito Bethencourt**
  - Principal Consultant at Codethink Ltd.
  - Codethink representative at CIP. Check [http://www.toscalix.com](http://www.toscalix.com)
Who are **Codethink**?

- Provide software engineering & consultancy services.
- Expert in Linux and Open Source software.
- Focus on embedded. Strong in automotive.
- UK Headquarters, serving clients in EU, US and Asia.
- Membership: CIP (founder member), OIN (2010), AGL (2015), GENIVI (2012)…
C.I.P.: a Linux Foundation Initiative

Provide a super long-term maintained industrial-grade embedded Linux platform.

Platinum Members

Silver Members

Open Source Summit Japan 2017
Talking points

1. Maintenance/support: the current picture.

2. Kernel maintenance strategies for industrial grade.

3. Limits to the maintenance lifetime.

4. How can CIP achieve a longer lifetime?
Maintenance/support: the current picture
Embedded, Mobile, Enterprise (aprox.)

Product lifetime

| Product | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 |
|---------|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|
| RHEL    | Development | Support (10 years) | Extended support |
| SLE     | Development | Support (10 years) | Extended support |
| Debian  | Development | Maintenance (3 years) | Extended support |
| Android | Development | Support (3 years) |
| LTS kernel(*) | Maintenance (6 years) |
| LSK(**) | 4m | 10m | 2m |
| LTSI(**) | 8m | Maintenance (2 years) |

(*) Assuming 2m on -next and 2m on mainline
(**) "On top" of the (LTS) Linux Kernel
CIP products

Railway Control System

- 3 – 5 years development time
- 2 – 4 years customer specific extensions
- 1 year initial safety certifications / authorization
- 3 – 6 months safety certifications / authorization for follow-up releases (depending on amount of changes)
- 25 – 50 years product lifetime
CIP products

Power Plant Control System

- 3 – 5 years development time
- 0.5 – 4 years customer specific extensions
- 6 - 8 years supply time.
- 15 years hardware maintenance after latest shipment
- 20 – 60 years product lifetime
CIP products vs other industries

CIP Products

Years of maintenance / support

Railway Control System
- Development (3-5 years)
- Customer extensions (3-4 years)
- Support (20-30 years)

Power Plant Control System
- Development (3-5 years)
- Customer extensions (3-4 years)
- Supply time-deployment (3-6 years)

Product

1 2 3 4 5 6 7 8 9

RHET
- Development
- Support

SLE
- Development

Debian
- Development

Android
- Development

LTS kernel(*)
- Main

LSK(**)
- 10m

LSI(**)
- Maintenance (4 years)

(*) Assuming 2m on next and 2m on m.
(**) "On top" of the (LTS) Linux Kernel

x 3 or 4
Codethink, are you sure you want to get into this? Really?
Kernel maintenance strategies for industrial grade.
Update!
Strategy 2

Extend current process:

SSSSSS... LTS
Strategy 3
Limits to the maintenance lifetime
Limits to lifetime - 2038

- Linux represents “wall-clock” time as number of seconds since 1970.
- With 32-bit signed values, maximum possible time is in 2038.
- On 32-bit architectures, time types in uAPI and many internal APIs are 32-bit.
- On all architectures, time types in some internal APIs and filesystem formats are 32-bit.
- Needs changes in kernel, libc, other libraries, some applications.
  - Kernel and GNU libc changes in progress.
- Probably not backport-able.
Limits to lifetime - hardware

- Support lifetime of most CPUs and SoCs is much less than 30 years
- Only most recent CIP kernel branch receives hardware support backports
- Some long-lived systems might require replacement of the Linux-based component - both hardware and kernel
Limits to lifetime - software

- Kernel internal APIs and their implementations change over time, sometimes dramatically
- Bug fixes may depend on those interface or implementation changes
- Backporting bug fixes from mainline to an older branch therefore becomes more difficult over time
How can CIP achieve a longer lifetime?
A longer lifetime - scope

- Scope of maintenance is based on needs of members
- Most architectures, drivers, filesystems, etc. are not used and their bugs can be ignored
- Greatly reduces effort to backport and review fixes when a branch is only maintained by CIP
A longer lifetime - fewer fixes needed

- Most important bugs are found and fixed quickly
- Number of bugs in a stable branch reduces over time, so does rate of fixes
- Difficulty of backporting fixes is counterbalanced by lower rate of fixes to handle
- Obscure bugs without security impact may not need fixing
A longer lifetime - collaboration

- CIP won’t be a vendor providing support to customers
- Members’ developers will take over maintenance, addressing own needs but sharing the work
- Lifetime of each kernel branch will be determined by the interest and capability of members
Thanks.

Questions?
Maintaining a Linux Kernel for 13 Years? You Must be Kidding Me. We Need at Least 30!