The kernel report

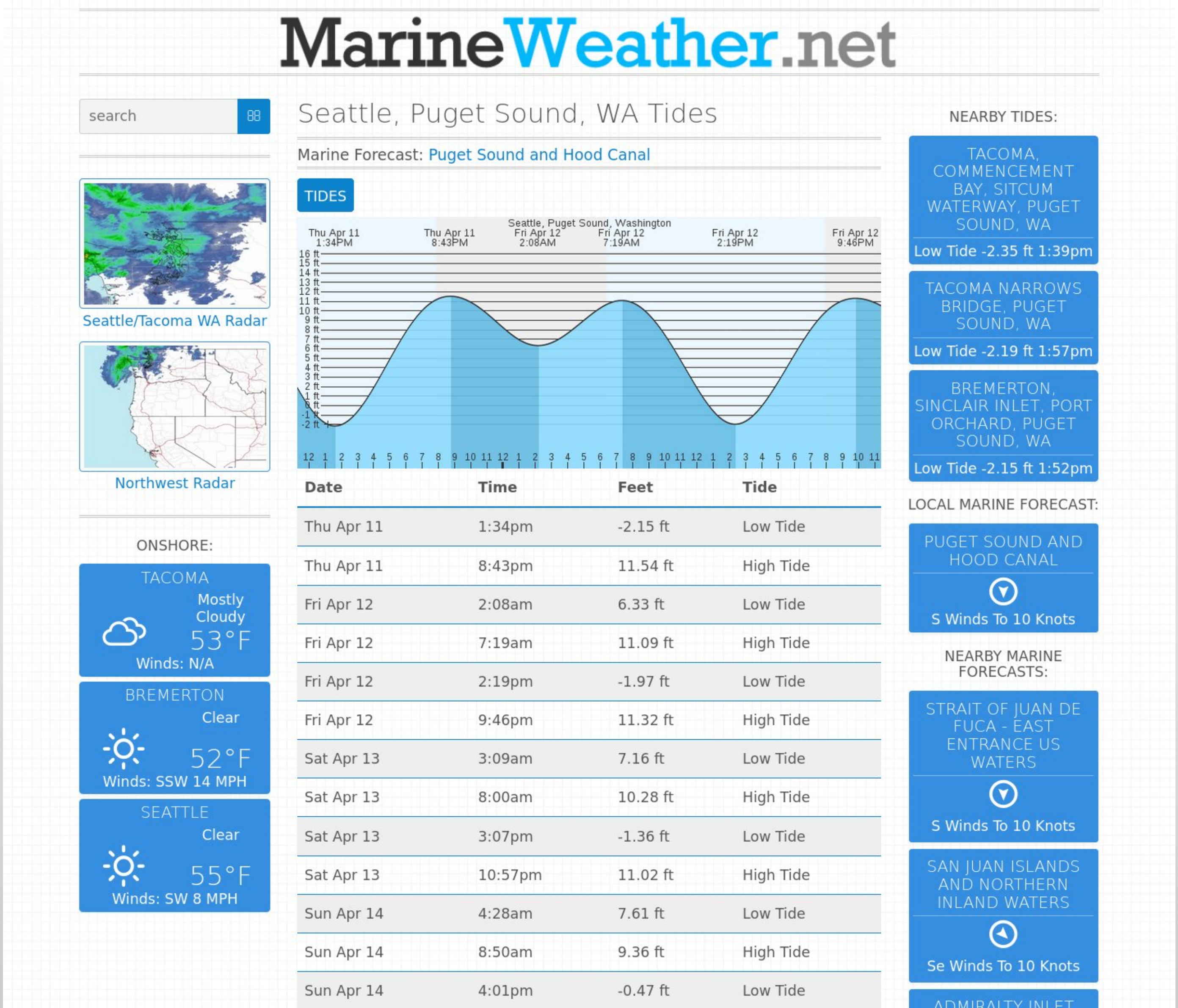
(OSS NA 2024 edition)

Jonathan Corbet
LWN.net
corbet@lwn.net

Part 1: Statistics

Recent release history

Release	Date	Commits	Devs	1st time
6.3	Apr 24	14,424	1,971	250
6.4	Jun 25	14,835	1,980	282
6.5	Aug 27	13,561	1,921	271
6.6	Oct 29	14,069	1,976	249
6.7	Jan 7	17,284	1,973	270
6.8	Mar 10	14,405	1,938	245



MarineWeather.net

search

Seattle/Tacoma WA Radar

Northwest Radar

ONSHORE:

TACOMA
Mostly Cloudy
53°F
Winds: N/A

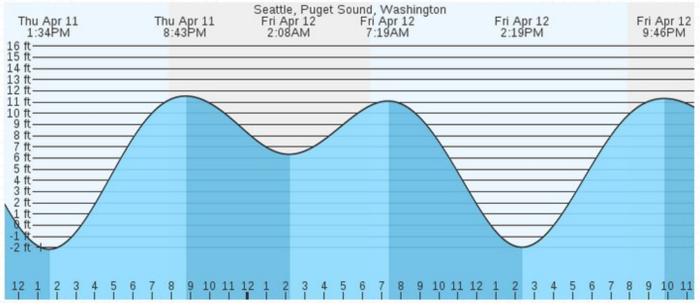
BREMERTON
Clear
52°F
Winds: SSW 14 MPH

SEATTLE
Clear
55°F
Winds: SW 8 MPH

Seattle, Puget Sound, WA Tides

Marine Forecast: Puget Sound and Hood Canal

TIDES



Date	Time	Feet	Tide
Thu Apr 11	1:34pm	-2.15 ft	Low Tide
Thu Apr 11	8:43pm	11.54 ft	High Tide
Fri Apr 12	2:08am	6.33 ft	Low Tide
Fri Apr 12	7:19am	11.09 ft	High Tide
Fri Apr 12	2:19pm	-1.97 ft	Low Tide
Fri Apr 12	9:46pm	11.32 ft	High Tide
Sat Apr 13	3:09am	7.16 ft	Low Tide
Sat Apr 13	8:00am	10.28 ft	High Tide
Sat Apr 13	3:07pm	-1.36 ft	Low Tide
Sat Apr 13	10:57pm	11.02 ft	High Tide
Sun Apr 14	4:28am	7.61 ft	Low Tide
Sun Apr 14	8:50am	9.36 ft	High Tide
Sun Apr 14	4:01pm	-0.47 ft	Low Tide

NEARBY TIDES:

TACOMA, COMMENCEMENT BAY, SITCUM WATERWAY, PUGET SOUND, WA
Low Tide -2.35 ft 1:39pm

TACOMA NARROWS BRIDGE, PUGET SOUND, WA
Low Tide -2.19 ft 1:57pm

BREMERTON, SINCLAIR INLET, PORT ORCHARD, PUGET SOUND, WA
Low Tide -2.15 ft 1:52pm

LOCAL MARINE FORECAST:

PUGET SOUND AND HOOD CANAL
S Winds To 10 Knots

NEARBY MARINE FORECASTS:

STRAIT OF JUAN DE FUCA - EAST ENTRANCE US WATERS
S Winds To 10 Knots

SAN JUAN ISLANDS AND NORTHERN INLAND WATERS
Se Winds To 10 Knots

ADMIRALTY INLET

Recent release history

Release	Date	Commits	Devs	1st time
6.3	Apr 24	14,424	1,971	250
6.4	Jun 25	14,835	1,980	282
6.5	Aug 27	13,561	1,921	271
6.6	Oct 29	14,069	1,976	249
6.7	Jan 7	**17,284**	1,973	270
6.8	Mar 10	14,405	1,938	245

Recent release history

Release	Date	Commits	Devs	1st time
6.3	Apr 24	14,424	1,971	250
6.4	Jun 25	14,835	1,980	282
6.5	Aug 27	13,561	1,921	271
6.6	Oct 29	14,069	1,976	249
6.7	Jan 7	17,284	1,973	270
6.8	Mar 10	14,405	1,938	245
6.9	**May 12?**	**12,766**	**1,761**	**220**

Coming in 6.9

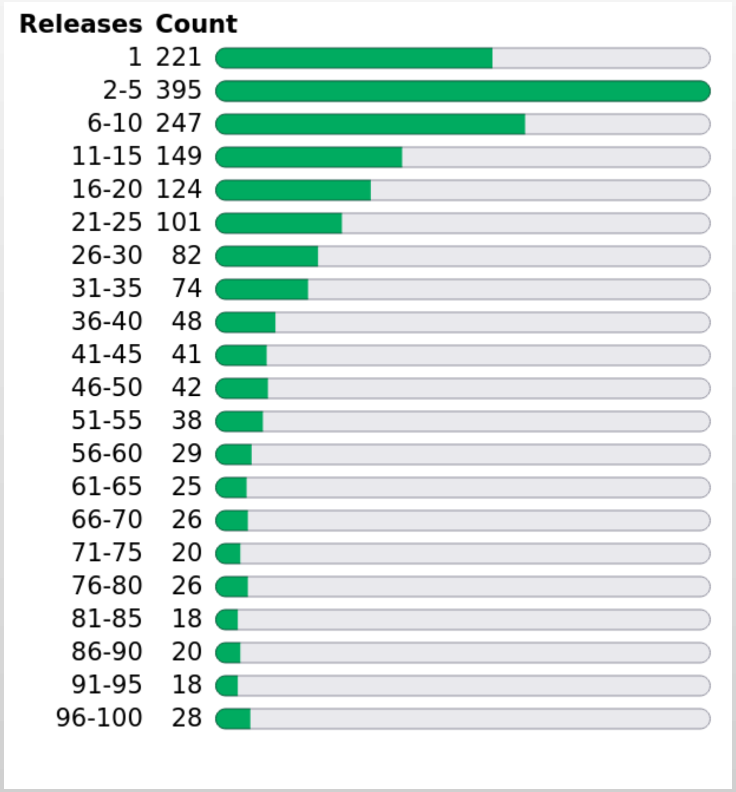
Intel FRED support
AMD SNP guests
pidfdfs
BPF arena
BPF token
Rust on arm64
Weighted interleaving

Contiguous PTE
DM virtual data opt.
XFS live repair
FUSE passthrough
Runtime energy model
Lots of device drivers
...

Recent release history

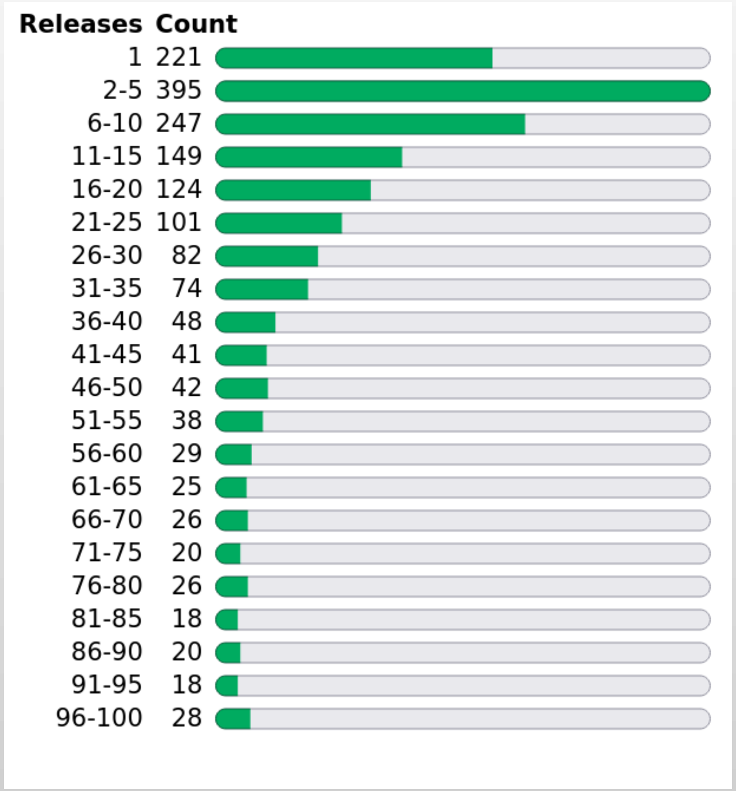
Release	Date	Commits	Devs	1st time
6.3	Apr 24	14,424	1,971	**250**
6.4	Jun 25	14,835	1,980	**282**
6.5	Aug 27	13,561	1,921	**271**
6.6	Oct 29	14,069	1,976	**249**
6.7	Jan 7	17,284	1,973	**270**
6.8	Mar 10	14,405	1,938	**245**
6.9	May 12?	12,766	1,761	**220**

Developer longevity

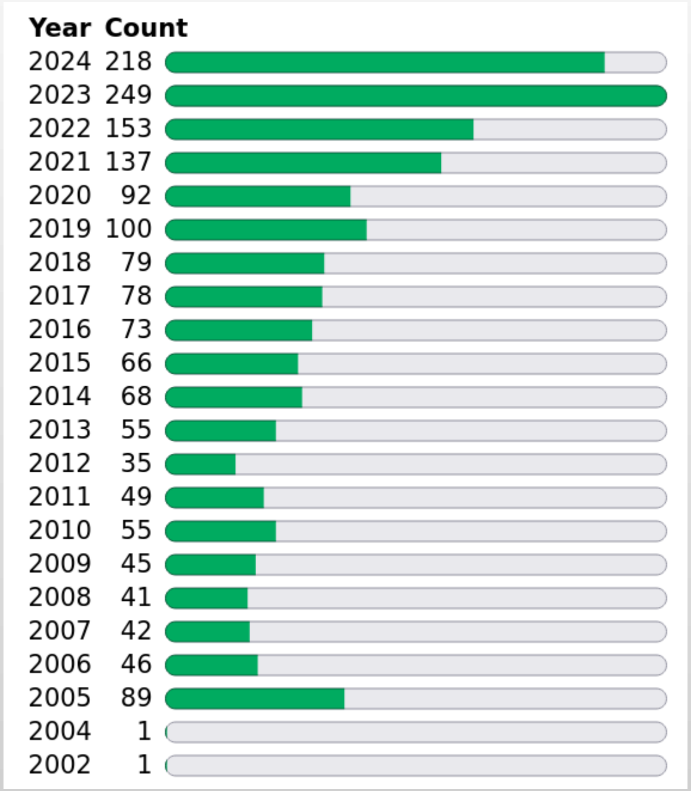


Releases	Count
1	221
2-5	395
6-10	247
11-15	149
16-20	124
21-25	101
26-30	82
31-35	74
36-40	48
41-45	41
46-50	42
51-55	38
56-60	29
61-65	25
66-70	26
71-75	20
76-80	26
81-85	18
86-90	20
91-95	18
96-100	28

Developer longevity



Releases	Count
1	221
2-5	395
6-10	247
11-15	149
16-20	124
21-25	101
26-30	82
31-35	74
36-40	48
41-45	41
46-50	42
51-55	38
56-60	29
61-65	25
66-70	26
71-75	20
76-80	26
81-85	18
86-90	20
91-95	18
96-100	28



Year	Count
2024	218
2023	249
2022	153
2021	137
2020	92
2019	100
2018	79
2017	78
2016	73
2015	66
2014	68
2013	55
2012	35
2011	49
2010	55
2009	45
2008	41
2007	42
2006	46
2005	89
2004	1
2002	1

Stable updates

Release	Updates	Commits
4.19	311	28,400
5.4	273	26,583
5.10	214	25,556
5.15	151	21,978
6.1	83	14,651
6.6	23	5,143

Security

CVE numbers

The problems with CVE numbers

A target in their own right

→ Bogus CVE-number problem

Many vulnerabilities never get CVEs

If you attempt to cherry-pick random patches you will NOT fix all of the known, and unknown, problems, but rather you will end up with a potentially more insecure system, and one that contains known bugs.
— Greg Kroah-Hartman

CVE numbers

(Mostly) ignored for years!

Certificate numbering authority

The body that issues CVEs for a project

Many projects have become CNAs

- curl
- GNU libc
- OpenNMS
- Apache
- Docker
- Document Foundation
- Kubernetes
- Python
- Debian
- ...

The kernel is now a CNA

Note, due to the layer at which the Linux kernel is in a system, almost any bug might be exploitable to compromise the security of the kernel, but the possibility of exploitation is often not evident when the bug is fixed. Because of this, the CVE assignment team is overly cautious and assign CVE numbers to any bugfix that they identify. This explains the seemingly large number of CVEs that are issued by the Linux kernel team.
— https://docs.kernel.org/process/cve.html

“Large number of CVEs”

→ About 800 assigned
...since late February

How to stay secure?

1) Attempt to track CVEs and backport fixes

2) Simply run the stable updates

As always, it is best to take all released kernel changes, as they are tested together in a unified whole by many community members, and not as individual cherry-picked changes.
— https://docs.kernel.org/process/cve.html

Stable updates

Release	Updates	Commits
4.19	311	28,400
5.4	273	26,583
5.10	214	25,556
5.15	151	21,978
6.1	83	14,651
6.6	23	5,143

It’s still the best answer we have.

The XZ backdoor

The XZ backdoor

This is **not** a kernel vulnerability!

```
+# Set XZ_VERSION (and LIBLZMA_VERSION). This is
needed to disable features
+# that aren't available in old XZ Utils versions.
+eval "$($XZ --robot --version)" || exit
```

https://lore.kernel.org/lkml/20240320183846.19475-12-lasse.collin@tukaani.org/

The XZ backdoor

This is **not** a kernel vulnerability!

...but could it be...?

The reality that we are struggling with is that the free software infrastructure on which much of computing runs is massively and painfully underfunded by society as a whole, and is almost entirely dependent on random people maintaining things in their free time because they find it fun, many of whom are close to burnout. This is, in many ways, the true root cause of this entire event.
— Russ Allbery

But kernel maintainers are paid!

Being maintainer feels like a punishment, and that cannot stand. We need help.
— Darrick Wong

Maintainers/longtime developers are burning out.
— Josef Bacik

But being a maintainer myself with a full-time job that is not to do my maintainership, I'm struggling to find time to work on this.
— Steve Rostedt

Dark areas in the kernel

Documentation
Build system
Many core-kernel areas
Drivers for older hardware
...

Dark areas in the kernel

Documentation
Build system
Many core-kernel areas
Drivers for older hardware
…
Maintainers

Kernel maintainership lacks support

Burned-out maintainers
Slowed development pace
Frustrated developers
Quality problems

Kernel maintainership lacks support

Burned-out maintainers
Slowed development pace
Frustrated developers
Quality problems

A security risk!

What’s to be done?

Be nice to maintainers

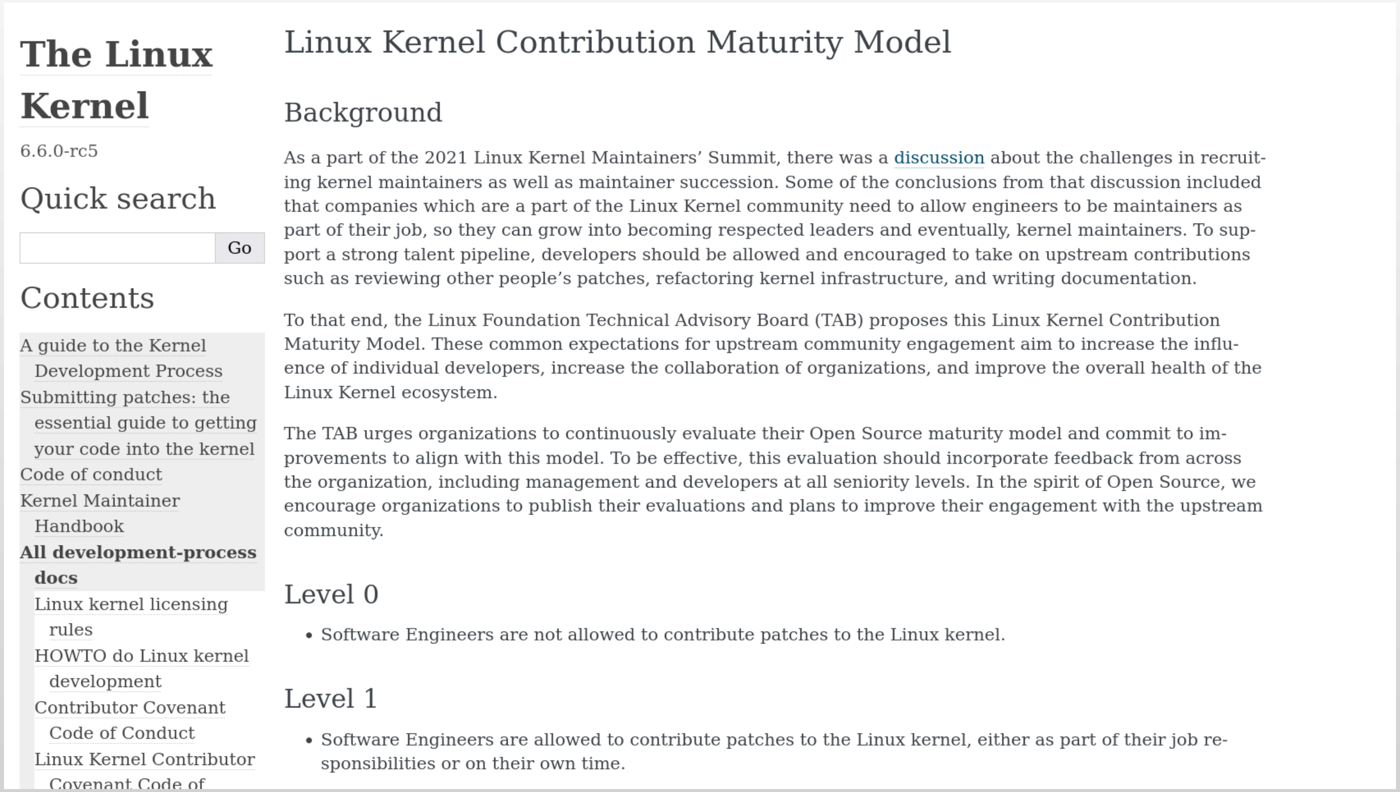
Help them to maintain their subsystem

- Review patches

Support the maintainer role

- as part of their job!

How does your company compare?



The Linux Kernel

6.6.0-rc5

Quick search

Go

Contents

Linux Kernel Contribution Maturity Model

Background

As a part of the 2021 Linux Kernel Maintainers' Summit, there was a discussion about the challenges in recruiting kernel maintainers as well as maintainer succession. Some of the conclusions from that discussion included that companies which are a part of the Linux Kernel community need to allow engineers to be maintainers as part of their job, so they can grow into becoming respected leaders and eventually, kernel maintainers. To support a strong talent pipeline, developers should be allowed and encouraged to take on upstream contributions such as reviewing other people's patches, refactoring kernel infrastructure, and writing documentation.

To that end, the Linux Foundation Technical Advisory Board (TAB) proposes this Linux Kernel Contribution Maturity Model. These common expectations for upstream community engagement aim to increase the influence of individual developers, increase the collaboration of organizations, and improve the overall health of the Linux Kernel ecosystem.

The TAB urges organizations to continuously evaluate their Open Source maturity model and commit to improvements to align with this model. To be effective, this evaluation should incorporate feedback from across the organization, including management and developers at all seniority levels. In the spirit of Open Source, we encourage organizations to publish their evaluations and plans to improve their engagement with the upstream community.

Level 0

- Software Engineers are not allowed to contribute patches to the Linux kernel.

Level 1

- Software Engineers are allowed to contribute patches to the Linux kernel, either as part of their job responsibilities or on their own time.

https://www.kernel.org/doc/html/latest/process/contribution-maturity-model.html

Open source is free like a puppy is free
— Scott McNealy

BPF

SPEAKEASY
PRODUCTIONS
eBPF
UNLOCKING THE KERNEL
PREMIERING NOV 8TH
AT KUBECON + CLOUNDNATIVECON CHICAGO
eBPF FOUNDATION
ISOVALENT
intel

Recent BPF work

BPF Tokens

BPF Arena

BPF work in progress

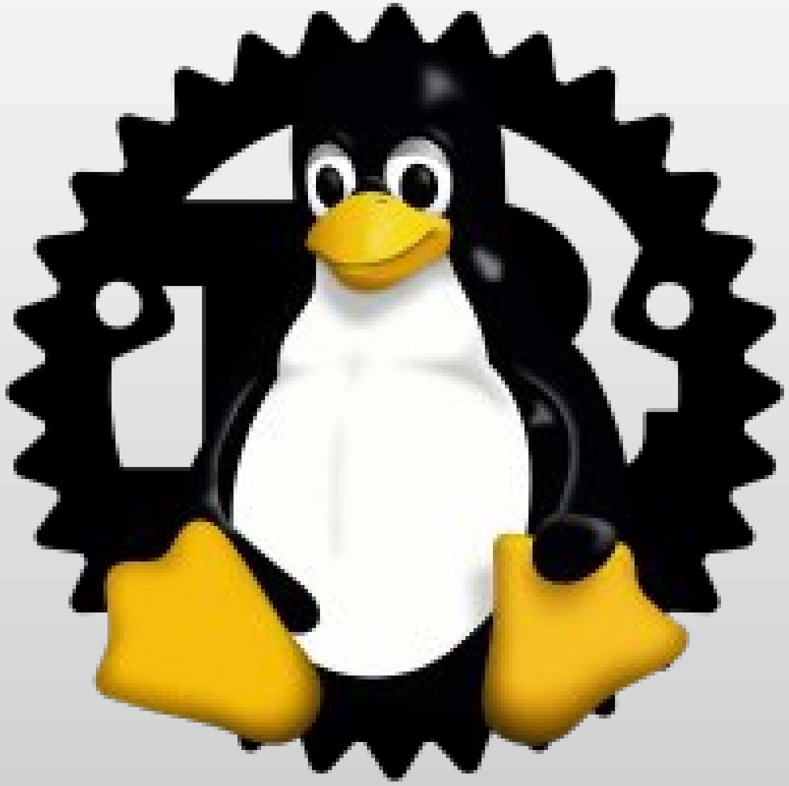
sched_ext.
Paravirt scheduling
FUSE-BPF
BPF network device
P4TC
Standardization

BPF challenges

Complexity

Resistance

Rust



Rust

A memory-safe language for the kernel

Rust status

Core support is maturing

Subsystem abstractions (slowly) added

A couple of sample drivers

Rust challenges

Winning over maintainers

Rust/C API correspondence

Getting abstractions upstream

Rust in general

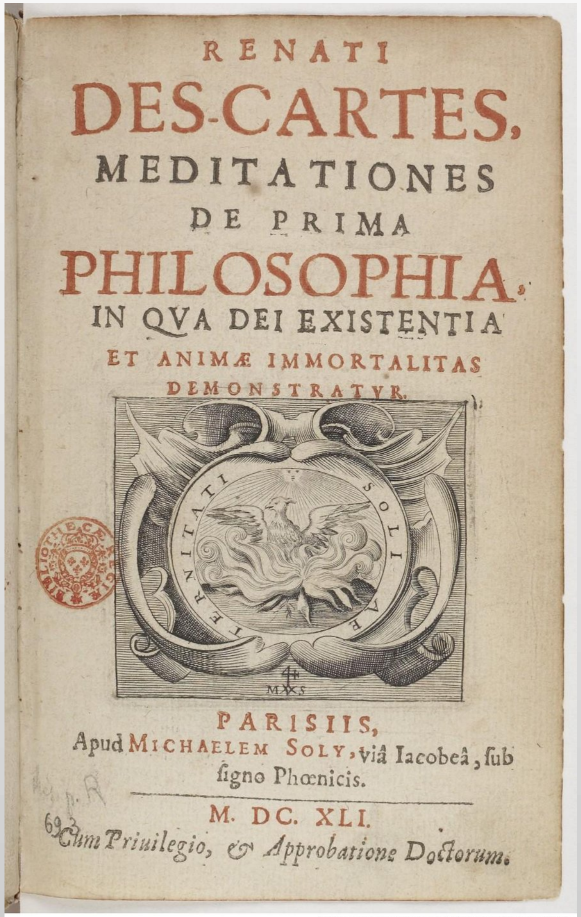
...so far, so good

Confidential computing

Confidential computing

Security in a hostile environment

What is true?


RENATI
DES-CARTES,
MEDITATIONES
DE PRIMA
PHILOSOPHIA,
IN QVA DEI EXISTENTIA
ET ANIMÆ IMMORTALITAS
DEMONSTRATVR.

PARISIIS,
Apud MICHAELEM SOLY, viâ Iacobeâ, ſub
ſigno Phœnicis.

M. DC. XLI.
Cum Priuilegio, & Approbatione Doctorum.

Source gallica.bnf.fr / Bibliothèque nationale de France

I suppose, accordingly, that everything that I see is false; I convince myself that nothing has ever existed of all that my deceitful memory recalls to me.
— Rene Descartes

I think, therefore I am

A modern cogito

“I am running on a secure CPU, therefore I am in control”.

Confidential computing

A secure, verified boot chain
CPU attestation
Encrypted memory

Confidential computing

A secure, verified boot chain
CPU attestation
Encrypted memory

AMD SEV+SNP
Intel TDX
Arm CCA

Confidential computing

A secure, verified boot chain
CPU attestation
Encrypted memory

→ Trust nothing!

Confidential computing

Is this plausible?

Are maintainers willing to try?

No time for...

EEVDF scheduler
Tasklets
IOCost
Kernel hardening
Memory tiering
mseal()
Kernel-text replication
Shadow stacks

Realtime preemption
Anonymous folios
Deadline servers
Stable kernel mgmt
Code tagging
composefs
Software interrupts
...

Questions?

(slides: https://lwn.net/talks/2024/kr-ossna.pdf)