

Recent Linux kernel development, "Quite well organized"

Early days of kernel development, "Highly fragmented"

How does community manage, "fragmented to stabilized migration"

The Evolution of Open Source: Navigating the Automotive Frontier

AGL All Member Meeting 2026-05 (Tokyo)

Hisao Munakata

hisao.munakata.vt(at)renesas.com

High Performance Computing Product Group
HPC SoC Business Division, SoC Software Enablement Department
Senior Director

2026-5-13

I say "Make or Buy or Collaborate", what does it mean?

Leading companies move from "make or buy" to "make or buy or collaborate"

- "Make" = Conduct in-house, closed-door R&D for differentiating technologies
- "Buy" = Source technologies externally for non-differentiating areas or technologies that require significant time
- "Collaborate" = Jointly develop the OSS components for differentiating technologies in partnership with other companies and communities
 - When the scale of development is large and the company cannot gather the necessary resources (funding, developers, etc.) on its own
 - By adopting an open approach, companies aim to ensure technology neutrality and accelerate development speed
 - Establish a de-facto by gaining a majority through cross company collaboration
 - Build sustainable solutions that can be maintained over the longterm
 - Avoid the risk of duplicate development among companies

"Make or Buy or Assemble", ... Really? Are you sure?

Make vs Buy. And now: Assemble. A 90-year-old decision with a third option

BUY

- Vendor scouting, PoCs, due diligence, security and legal reviews
- Price and SLA negotiations, IP/indemnity, compliance
- Difficulty of writing long-term specs for evolving needs

MAKE

- Staffing, project management, knowledge retention
- Integration, testing, certification, sustaining engineering
- Full ownership of roadmap and liability

ASSEMBLE (OSS)

- Integration, qualification, toolchain setup, compliance
- Fork and backport strategy, long-term maintenance
- Security operations ownership: CVE triage, regression, validation, rollout

For each component: which option minimizes marginal cost at the required assurance level?

Based on Ronald Coase, "The Nature of the Firm" (1937): firms internalize until internal coordination cost equals market transaction cost

08 March 2026 | Copyright ©2026 COVESA

6



<https://x.gd/dR60E>

- The automotive industry has not yet reached the point where it recognizes that "business cannot function without OSS," and it appears to treat OSS merely as one "option" among others, on par with other commercial software products. This is likely why a "wait-and-see" attitude remains the prevailing view.
- In other industries (such as IT and manufacturing), OSS is viewed as a "critical business strategy factor," and companies are actively embracing it. AGL has a role to play in raising awareness of this across the entire automotive industry.

Recent Linux kernel development, "Quite well organized"

Early days of kernel development, "Highly fragmented"

How does community manage, "fragmented to stabilized migration"

Linux kernel 7.0

kernel development flow (now)

Each series release record (v3, v5, v6)

Recent Linux kernel development "Quite well organized"

Linux "kernel 7" is not a product name, but just a version number

Protocol	Location
HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/

Latest Release

7.0.6 

mainline:	7.1-rc3	2026-05-10	[tarball]	[patch]	[inc. patch]	[view diff]	[browse]
stable:	7.0.6	2026-05-11	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
stable:	6.19.14 [EOL]	2026-04-22	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	6.18.29	2026-05-11	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	6.12.87	2026-05-08	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	6.6.138	2026-05-08	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	6.1.172	2026-05-08	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	5.15.206	2026-05-08	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
longterm:	5.10.255	2026-05-08	[tarball]	[pgp]	[patch]	[inc. patch]	[view diff] [browse] [changelog]
linux-next:	next-20260508	2026-05-08					[browse]

<https://www.kernel.org/>

Recent Linux kernel development, "Quite well organized"

Early days of kernel development, "Highly fragmented"

How does community manage, "fragmented to stabilized migration"

Linux kernel 7.0

kernel development flow (now)

Each series release record (v3, v5, v6)

Conversely, "Windows 7" was product name, its version was NT6.1

Product Name	SW version	Release Date	Release interval (Major)
Windows 1.0	1.0	1985/11/1	
Windows 2.0	2.0	1987/12/1	2.0 years
Windows 3.0	3.0	1990/5/1	2.5 years
Windows 3.1	3.1	1992/4/1	2.0 years
Windows 95	4.0	1995/8/1	3.3 years
Windows 98	4.1	1998/6/1	2.8 years
Windows Me	4.9	2000/9/1	2.3 years
Windows NT 3.1	NT 3.1	1993/7/1	(NT series starts)
Windows NT 4.0	NT 4.0	1996/8/1	3.0 years
Windows 2000	NT 5.0	2000/2/1	3.5 years
Windows XP	NT 5.1	2001/10/1	1.7 years
Windows Vista	NT 6.0	2007/1/1	5.3 years
Windows 7	NT 6.1	2009/10/1	2.8 years
Windows 8	NT 6.2	2012/10/1	3.0 years
Windows 10	NT 10.0	2015/7/1	1.8 years
Windows 11	NT 10.0	2021/10/1	6.2 years

Windows の情報

エディション	Windows 11 Home
バージョン	25H2
インストール日	2026/02/06
OS ビルド	26200.8246
イクスペリエンス	Windows 機能イクスペリエンス パック 1000.26100.297.0

[Microsoft サービス規約](#)
[Microsoft ソフトウェアライセンス条項](#)

6.19 to 7.0 migration does not mean a major feature change/drop

LKML.ORG

Messages in this thread

- *First message in thread*
- **Linus Torvalds**

[lkml] [2026] [Feb] [22] [last100] [RSS](#)

Views: [wrap] [headers] [forward]

From Linus Torvalds <>
Date Sun, 22 Feb 2026 14:40:09 -0800
Subject Linux 7.0-rc1

You all know the drill by now: two weeks have passed, and the kernel merge window is closed.

We have a new major number purely because I'm easily confused and not good with big numbers.

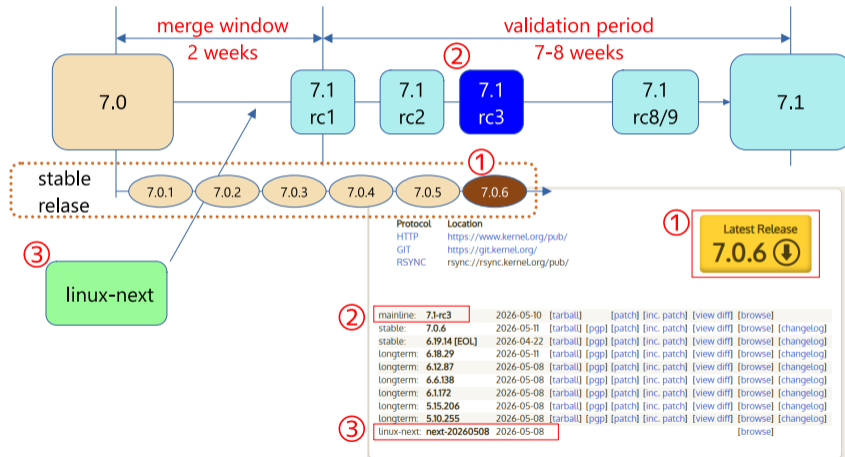
We haven't done releases based on features (or on "stable vs unstable") for a long long time now. So that new major number does **not** mean that we have some big new exciting feature, or that we're somehow leaving old interfaces behind. It's the usual "solid progress" marker, nothing more.

With our normal release schedule of 5-6 releases per year and my antipathy to big version numbers, you should basically expect us to bump the major number roughly every 3.5 years.

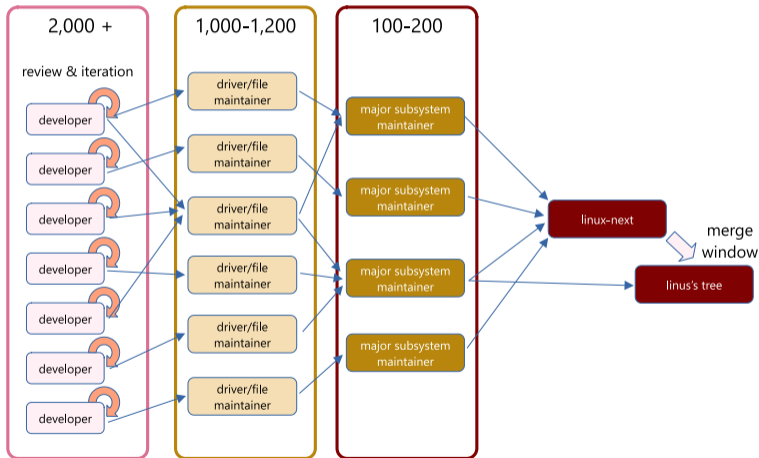
And yeah, I don't have a solid plan for when the major number itself gets big. But doing the math - by that time, I expect that we'll have somebody more competent in charge who isn't afraid of numbers past the teens. So I'm not going to worry about it.

<https://lkml.org/lkml/2026/2/22/297>

The current kernel development flow (established after v3 series)



The current "proven" patch submission flow



kernel v6 to v7 (release interval : $ave=64.3$ 、 $\sigma=2.83$)

Release	Date	Duration	RCs	Developer	Commits		
					Total	Reg.	Merge
v7.1			2	2,066	14,438	13,417	1,021
v7.0	2026-04-12	63	7	2,362	15,623	14,251	1,372
v6.19	2026-02-08	70	8	2,137	15,656	14,344	1,312
v6.18	2025-11-30	63	7	2,134	15,035	13,710	1,325
v6.17	2025-09-28	63	7	2,038	14,334	13,089	1,245
v6.16	2025-07-27	63	7	2,056	15,924	14,639	1,285
v6.15	2025-05-25	62	7	2,066	15,942	14,612	1,330
v6.14	2025-03-24	64	7	1,898	12,114	11,003	1,111
v6.13	2025-01-19	63	7	2,002	14,168	12,982	1,186
v6.12	2024-11-17	63	7	2,074	14,607	13,344	1,263
v6.11	2024-09-15	63	7	1,969	15,122	13,890	1,232
v6.10	2024-07-14	63	7	1,917	14,561	13,312	1,249
v6.9	2024-05-12	63	7	2,025	15,678	14,480	1,198
v6.8	2024-03-10	63	7	1,938	15,639	14,405	1,234
v6.7	2024-01-07	70	8	1,973	18,404	17,284	1,120
v6.6	2023-10-29	63	7	1,976	15,289	14,069	1,220
v6.5	2023-08-27	63	7	1,920	14,674	13,561	1,113
v6.4	2023-06-25	63	7	1,979	16,011	14,835	1,176
v6.3	2023-04-23	63	7	1,970	15,637	14,424	1,213
v6.2	2023-02-19	70	8	2,090	16,840	15,536	1,304
v6.1	2022-12-11	70	8	2,045	15,109	13,942	1,167
v6.0	2022-10-02	63	7	2,033	16,584	15,402	1,182

3.5 years
1,287 dayslongterm
(LTS) version

Recent Linux kernel development, "Quite well organized"

Early days of kernel development, "Highly fragmented"

How does community manage, "fragmented to stabilized migration"

Linux kernel 7.0

kernel development flow (now)

Each series release record (v3, v5, v6)

kernel v5 series (release interval : ave=65.8、 $\sigma=5.25$)

Release	Date	Duration	RCs	Developer	Commits		
					Total	Reg.	Merge
v5.19	2022-07-31	70	8	2,085	16,399	15,134	1,265
v5.18	2022-05-22	63	7	2,022	16,205	14,954	1,251
v5.17	2022-03-20	70	8	1,899	14,199	13,038	1,161
v5.16	2022-01-09	70	8	1,989	15,384	14,190	1,194
v5.15	2021-10-31	63	7	1,797	13,473	12,377	1,096
v5.14	2021-08-29	63	7	1,913	15,871	14,735	1,136
v5.13	2021-06-27	63	7	2,064	17,187	16,030	1,157
v5.12	2021-04-25	70	8	1,872	14,167	13,015	1,152
v5.11	2021-02-14	63	7	1,913	15,477	14,340	1,137
v5.10	2020-12-13	63	7	1,970	17,467	16,175	1,292
v5.9	2020-10-11	70	8	1,913	16,062	14,858	1,204
v5.8	2020-08-02	63	7	1,990	17,595	16,306	1,289
v5.7	2020-05-31	63	7	1,878	15,025	13,901	1,124
v5.6	2020-03-29	63	7	1,711	13,691	12,665	1,026
v5.5	2020-01-26	63	7	1,883	15,497	14,350	1,147
v5.4	2019-11-24	70	8	1,876	15,725	14,619	1,106
v5.3	2019-09-15	70	8	1,882	15,784	14,605	1,179
v5.2	2019-07-07	75	7	1,798	15,295	14,194	1,101
v5.1	2019-04-23	51	7	1,726	13,954	12,864	1,090
v5.0	2019-03-03	70	8	1,757	13,921	12,808	1,113

3.6 years
1,308 days

Recent Linux kernel development, "Quite well organized"

Early days of kernel development, "Highly fragmented"

How does community manage, "fragmented to stabilized migration"

Linux kernel 7.0

kernel development flow (now)

Each series release record (v3, v5, v6)

kernel v4 series (release interval : ave=67.3、 $\sigma=4.22$)

Release	Date	Duration	RCs	Developer	Commits		
					Total	Reg.	Merge
v4.20	2018-12-23	62	7	1,748	14,995	13,884	1,111
v4.19	2018-10-22	71	8	1,749	15,204	14,043	1,161
v4.18	2018-08-12	70	8	1,731	14,432	13,283	1,149
v4.17	2018-06-03	63	7	1,712	14,745	13,541	1,204
v4.16	2018-04-01	63	7	1,803	14,896	13,630	1,266
v4.15	2018-01-28	77	9	1,802	16,223	14,866	1,357
v4.14	2017-11-12	70	8	1,755	14,659	13,452	1,207
v4.13	2017-09-03	63	7	1,679	14,150	13,006	1,144
v4.12	2017-07-02	63	7	1,820	15,736	14,570	1,166
v4.11	2017-04-30	70	8	1,744	13,891	12,724	1,167
v4.10	2017-02-19	70	8	1,678	14,249	13,029	1,220
v4.9	2016-12-11	70	8	1,728	17,392	16,214	1,178
v4.8	2016-10-02	70	8	1,595	14,552	13,382	1,170
v4.7	2016-07-24	70	7	1,580	13,433	12,283	1,150
v4.6	2016-05-15	63	7	1,679	14,618	13,517	1,101
v4.5	2016-03-13	63	7	1,539	13,173	12,080	1,093
v4.4	2016-01-10	70	8	1,577	14,082	13,071	1,011
v4.3	2015-11-01	63	7	1,624	13,282	12,274	1,008
v4.2	2015-08-30	70	8	1,596	14,750	13,694	1,056
v4.1	2015-06-21	70	8	1,537	12,965	11,916	1,049
v4.0	2015-04-12	63	7	1,457	11,306	10,346	960

3.9 years
1,420 days

Meltdown
Spectre

Recent Linux kernel development, "Quite well organized"

Early days of kernel development, "Highly fragmented"

How does community manage, "fragmented to stabilized migration"

Linux kernel 7.0

kernel development flow (now)

Each series release record (v3, v5, v6)

kernel v3 series (release interval : ave=80.5、 $\sigma=7.71$)

Release	Date	Duration	RCs	Developer	Commits		
					Total	Reg.	Merge
v3.19	2015-02-08	63	7	1,454	13,652	12,617	1,035
v3.18	2014-12-07	63	7	1,457	12,361	11,379	982
v3.17	2014-10-05	63	7	1,436	13,331	12,354	977
v3.16	2014-08-03	56	7	1,478	13,830	12,804	1,026
v3.15	2014-06-08	70	8	1,496	14,871	13,722	1,149
v3.14	2014-03-30	70	8	1,444	13,415	12,311	1,104
v3.13	2014-01-19	77	8	1,362	13,288	12,127	1,161
v3.12	2013-11-03	62	7	1,335	11,906	10,927	979
v3.11	2013-09-02	64	7	1,270	11,851	10,893	958
v3.10	2013-06-30	63	7	1,391	14,737	13,637	1,100
v3.9	2013-04-28	69	8	1,394	12,956	11,910	1,046
v3.8	2013-02-18	70	7	1,262	13,526	12,394	1,132
v3.7	2012-12-10	71	8	1,279	13,005	11,990	1,015
v3.6	2012-09-30	71	7	1,225	11,109	10,226	883
v3.5	2012-07-21	62	7	1,200	11,793	10,978	815
v3.4	2012-05-20	63	7	1,293	11,829	10,899	930
v3.3	2012-03-18	74	7	1,246	11,416	10,550	866
v3.2	2012-01-04	72	7	1,317	12,695	11,881	814
v3.1	2011-10-24	95	10	1,167	9,380	8,693	687
v3.0	2011-07-21	64	7	1,135	9,843	9,153	690

3.7 years
1,360 days

kernel.org
Server hacked

Recent Linux kernel development, "Quite well organized"

Early days of kernel development, "Highly fragmented"

How does community manage, "fragmented to stabilized migration"

Look into wild days (2.4 to 2.6)

Each series release record (2.4, 2.5, 2.6)

Things happened during wild days

Early days of kernel development "Highly fragmented"

kernel major number migration history : two different phases

Version	Date	LOC	Reason
6.19 → 7.0	2026-04-12	40,000k	Not related to major feature update Not include major architectural change
4.20 → 5.0	2019-03-03	27,000k	
3.19 → 4.0	2015-04-12	19,500k	
2.6.39 → 3.0	2011-07-21	12,000k	
* 2.4.23/ 2.5.75 → 2.6	2003-12-17	6,000k	SMP(16-32 core), USB, ISA Plug&Play
2,2,17/ 2.3.99 → 2.4	2001-01-04	3,000k	SMP(4 core), PowerPC/Alpha/Spark64
2.0.36/ 2.1.132 → 2.2	1999-01-20	1,500k – 1,800k	Network support (VFS, IPv6, Firewall)
1.2.13/ 1.3.100 → 2.0	1996-06-09	450k – 500k	SMP initial,
1.0	1994-03-14	170k	1 st official release, TCP/IP, File System
0.1	1991-08-25	10k	"just a hobby, won't be big and professional like GNU"

1.3 / 2.1 / 2.3 / 2.5 series are **DEVELOPMENT** branch

1.2 / 2.2 / 2.4 / 2.6 series are **STABLE** branch

* kernel 2.4 series development continues even after 2.6 development start, till 2.6.28 release time (2008-12)
And, kept maintained till 2.4.27.11 (2011-12-18)

Recent Linux kernel development, "Quite well organized"

Early days of kernel development, "Highly fragmented"

How does community manage, "fragmented to stabilized migration"

Look into wild days (2.4 to 2.6)

Each series release record (2.4, 2.5, 2.6)

Things happened during wild days

2.4/2.6 series development was completely different from now

Series	Start	End	Duration		Release number	Release interval	
			days	years		Days	σ
6.0 series	2022-10-02	2026-04-12	1,288	(3.53)	19	64.3	2.83
5.0 series	2019-03-03	2022-10-01	1,308	(3.58)	19	65.8	5.25
4.0 series	2015-04-12	2019-03-02	1,420	(3.89)	20	67.3	4.22
3.0 series	2011-07-21	2015-04-11	1,360	(3.73)	19	80.5	7.71
2.6 series	2003-12-18	2011-07-20	2,771	(7.59)	39	69.4	23.6
2.4 series	2001-01-04	2008-12-01	2,888	(7.91)	37	78.1	79.19



Recent Linux kernel development, "Quite well organized"

Early days of kernel development, "Highly fragmented"

How does community manage, "fragmented to stabilized migration"

Look into wild days (2.4 to 2.6)

Each series release record (2.4, 2.5, 2.6)

Things happened during wild days

kernel v2.6 series late (release interval : ave=81.6、σ=9.84)

Release	Date	Duration	RCs	Developer	Commits		
					Total	Reg.	Merge
v2.6.39	2011-05-18	65	7	1,256	11,031	10,269	762
v2.6.38	2011-03-14	69	8	1,199	10,393	9,577	816
v2.6.37	2011-01-04	76	8	1,285	12,207	11,446	761
v2.6.36	2010-10-20	80	8	1,185	10,173	9,501	672
v2.6.35	2010-08-01	77	6	1,193	10,487	9,801	686
v2.6.34	2010-05-16	81	7	1,154	10,133	9,443	690
v2.6.33	2010-02-24	84	8	1,196	11,684	10,871	813
v2.6.32	2009-12-02	84	7	1,250	11,806	10,998	808
v2.6.31	2009-09-09	92	9	1,169	11,713	10,883	830
v2.6.30	2009-06-09	78	8	1,152	13,008	11,989	1,019
v2.6.29	2009-03-23	89	8	1,183	12,526	11,718	808
v2.6.28	2008-12-24	76	9	1,085	9,784	9,048	736
v2.6.27	2008-10-09	88	9	1,026	11,371	10,628	743
v2.6.26	2008-07-13	88	9	1,031	10,487	9,941	546
v2.6.25	2008-04-16	83	9	1,131	12,707	12,243	464
v2.6.24	2008-01-24	107	8	984	10,353	9,836	517
v2.6.23	2007-10-09	93	9	920	7,075	6,662	413
v2.6.22	2007-07-08	74	7	878	6,840	6,526	314
v2.6.21	2007-04-25	80	7	773	5,325	4,992	333
v2.6.20	2007-02-04	67	7	672	4,947	4,733	214

7.6 years
2,771 days

v2.6 series total

kernel v2.6 series early (release interval : $ave=56.7$, $\sigma=27.21$)

Release	Date	Duration	RCs	Developer	Commits		
					Total	Reg.	Merge
v2.6.19	2006-11-29	71	6	815	7,133	6,744	389
v2.6.18	2006-09-19	94	7	821	6,791	6,323	468
v2.6.17	2006-06-17	90	6	733	6,113	5,727	386
v2.6.16	2006-03-19	76	6	718	5,729	5,366	363
v2.6.15	2006-01-02	67	7	621	5,244	4,823	421
v2.6.14	2005-10-27	60	5	562	4,101	3,766	335
v2.6.13	2005-08-28	72	7	539	4,017	3,747	270
v2.6.12	2005-06-17	107	5	375	2,017	1,885	132
v2.6.11	2005-03-02	68	6-7	Minimal stabilization period			
v2.6.10	2004-12-24	66	6				
v2.6.9	2004-10-19	66	5-6				
v2.6.8	2004-08-14	59	4-5				
v2.6.7	2004-06-16	37	2				
v2.6.6	2004-05-10	36	2				
v2.6.5	2004-04-04	24	1				
v2.6.4	2004-03-11	22	0-1				
v2.6.3	2004-02-18	14	0				
v2.6.2	2004-02-04	26	1				
v2.6.1	2004-01-09	22	0-1				
v2.6.0	2003-12-18	-10	6-7				

Start using
"git"

Minimal
stabilization
period

kernel v2.4 series (release interval : $ave=78.1$, $\sigma=79.19$)

Release	Date	Duration
2.4.37	2008-12-02	336
2.4.36	2008-01-01	159
2.4.35	2007-07-26	215
2.4.34	2006-12-23	134
2.4.33	2006-08-11	268
2.4.32	2005-11-16	168
2.4.31	2005-06-01	58
2.4.30	2005-04-04	75
2.4.29	2005-01-19	63
2.4.28	2004-11-17	102
2.4.27	2004-08-07	115
2.4.26	2004-04-14	56
2.4.25	2004-02-18	44
2.4.24	2004-01-05	38
2.4.23	2003-11-28	95
2.4.22	2003-08-25	73
2.4.21	2003-06-13	197
2.4.20	2002-11-28	117
2.4.19	2002-08-03	159

Release	Date	Duration
2.4.18	2002-02-25	66
2.4.17	2001-12-21	25
2.4.16	2001-11-26	4
2.4.15	2001-11-22	17
2.4.14	2001-11-05	12
2.4.13	2001-10-24	13
2.4.12	2001-10-11	2
2.4.11	2001-10-09	16
2.4.10	2001-09-23	38
2.4.9	2001-08-16	5
2.4.8	2001-08-11	22
2.4.7	2001-07-20	17
2.4.6	2001-07-03	38
2.4.5	2001-05-26	28
2.4.4	2001-04-28	29
2.4.3	2001-03-30	36
2.4.2	2001-02-22	23
2.4.1	2001-01-30	26
2.4.0	2001-01-04	

7.9 years
2,889 days

The final 2.4.37.11
stable release was
on 2011-12-18

Embedded loved 2.4
series kernel because
2.6 series are

- Too big (RAM/ROM)
- Slow boot
- Too complicated (VM, scheduler,..)
- Move too fast
- Need NOR boot
- non-MMU support

Recent Linux kernel development, "Quite organized"

Early days of kernel development, "Highly fragmented"

How does community manage, "fragmented to stabilized migration"

Look into into wild days (2.4 to 2.6)

Each series release record (2.4, 2.5, 2.6)

Things happened during wild days

Development branch (odd number) is almost "weekly snapshot"

Release	Date	Duration
2.5.83	2003-11-09	15
2.5.82	2003-10-25	6
2.5.81	2003-10-19	14
2.5.80	2003-10-05	14
2.5.79	2003-09-21	13
2.5.78	2003-09-08	17
2.5.77	2003-08-22	14
2.5.76	2003-08-08	19
2.5.75	2003-07-20	7
2.5.74	2003-07-13	8
2.5.73	2003-07-05	10
2.5.72	2003-06-25	11
2.5.71	2003-06-14	6
2.5.70	2003-06-08	8
2.5.69	2003-05-31	13
2.5.68	2003-05-18	10
2.5.67	2003-05-08	7
2.5.66	2003-05-01	10
2.5.65	2003-04-21	7
2.5.64	2003-04-14	3
2.5.63	2003-04-11	10
2.5.62	2003-04-01	7
2.5.61	2003-03-25	6
2.5.60	2003-03-19	10
2.5.59	2003-03-09	17
2.5.58	2003-02-20	12
2.5.57	2003-02-08	10
2.5.56	2003-01-29	10
2.5.55	2003-01-19	8

Release	Date	Duration
2.5.54	2003-01-11	6
2.5.53	2003-01-05	18
2.5.52	2002-12-18	3
2.5.51	2002-12-15	14
2.5.50	2002-12-01	3
2.5.49	2002-11-28	11
2.5.48	2002-11-17	6
2.5.47	2002-11-11	7
2.5.46	2002-11-04	4
2.5.45	2002-10-31	10
2.5.44	2002-10-21	2
2.5.43	2002-10-19	5
2.5.42	2002-10-14	8
2.5.41	2002-10-06	5
2.5.40	2002-10-01	9
2.5.39	2002-10-10	6
2.5.38	2002-10-04	4
2.5.37	2002-09-30	7
2.5.36	2002-09-23	7
2.5.35	2002-09-16	7
2.5.34	2002-09-09	4
2.5.33	2002-09-05	10
2.5.32	2002-08-26	10
2.5.31	2002-08-16	7
2.5.30	2002-08-09	6
2.5.29	2002-08-03	5
2.5.28	2002-07-29	10
2.5.27	2002-07-19	5
2.5.26	2002-07-14	13

Release	Date	Duration
2.5.25	2002-07-01	15
2.5.24	2002-06-16	6
2.5.23	2002-06-10	10
2.5.22	2002-05-31	11
2.5.21	2002-05-20	8
2.5.20	2002-05-12	11
2.5.19	2002-05-01	10
2.5.18	2002-04-21	4
2.5.17	2002-04-17	3
2.5.16	2002-04-14	13
2.5.15	2002-04-01	8
2.5.14	2002-03-24	7
2.5.13	2002-03-17	7
2.5.12	2002-03-10	6
2.5.11	2002-03-04	8
2.5.10	2002-02-24	4
2.5.9	2002-02-20	9
2.5.8	2002-02-11	3
2.5.7	2002-02-08	4
2.5.6	2002-02-04	31
2.5.5	2002-01-04	14
2.5.4	2001-12-21	4
2.5.3	2001-12-17	7
2.5.2	2001-12-10	10
2.5.1	2001-11-30	7
2.5.0	2001-11-23	

Things happened during the 2.4/2.5 "wild development" period

Multiple huge changes went in parallel during that period

- O(1) scheduler
- VM rewrite
- preempt
- block layer rewrite
- driver model
- kobject/sysfs
- module changes

Fundamental core architectural changes occurred

Automotive industry may be in the similar situation NOW

SDV forces people to adopt and integrate various new software technologies

- ECU layout reconsideration (split to consolidate)
- End-to-end autonomous driving (NOA)
- Cloud resource integration (Container orchestration)
- Cybersecurity management
- Use of a virtualized environment (Hypervisor)
- SW update (SOTA/FOTA)
- CAN to LAN migration

We should learn from Linux wild-days development history

Recent Linux kernel development, "Quite well organized"

Early days of kernel development, "Highly fragmented"

How does community manage, "fragmented to stabilized migration"

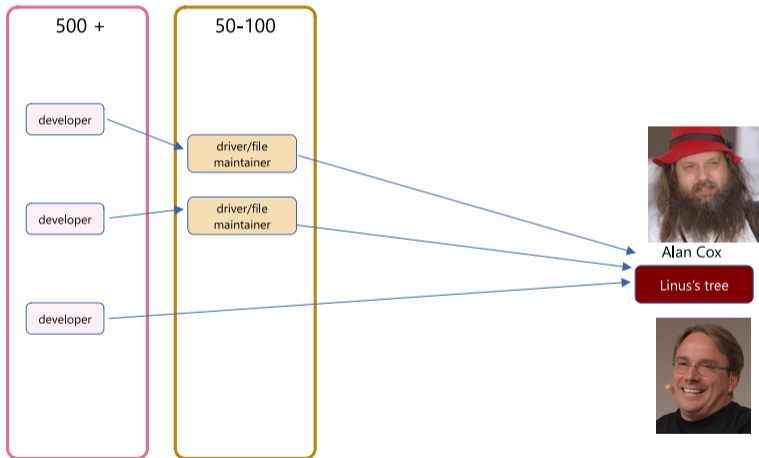
How does community manage "fragmented to stabilized migration"

Recent Linux kernel development, "Quite well organized"

Early days of kernel development, "Highly fragmented"

How does community manage, "fragmented to stabilized migration"

Early days (until 2003.3) patch flow (almost direct to Linus)

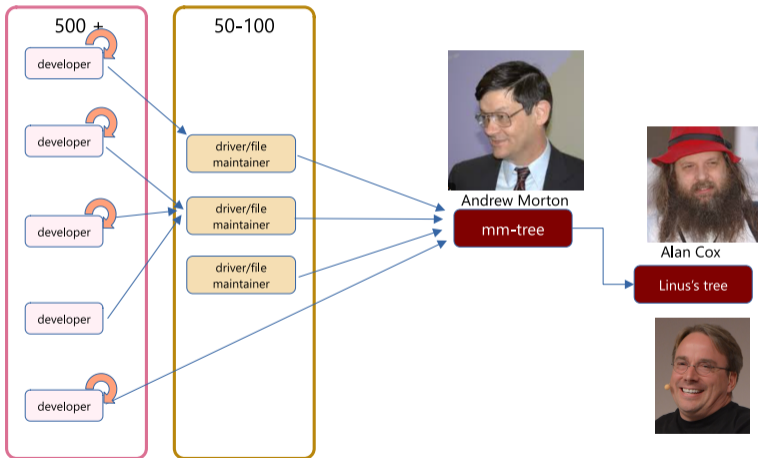


Recent Linux kernel development, "Quite well organized"

Early days of kernel development, "Highly fragmented"

How does community manage, "fragmented to stabilized migration"

mm-tree started working as "pre-validation gate" (after 2003.3)

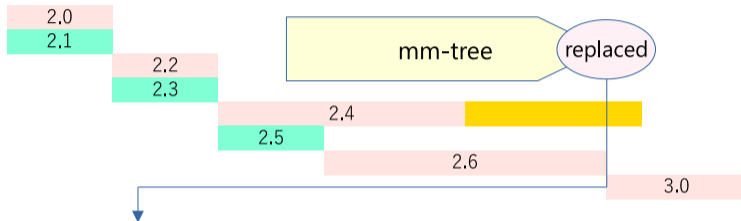


Recent Linux kernel development, "Quite well organized"

Early days of kernel development, "Highly fragmented"

How does community manage, "fragmented to stabilized migration"

How kernel development achieved long-term stability



"git" realized distributed development (= local mm)

rc-process (merge-period / validation-period) established

IT-assisted automated validation became available (= CI/CD)

Moved to time-based (not feature-based) periodic release

Kernel development statistics (kernel 7.0)

kernel 7.0 (release 2026-04-12) contributors

Most active 7.0 employers				Most active 7.0 developers							
By changesets		By lines changed		By changesets		By changed lines					
(Unknown)	1666	11.7%	AMD	139764	17.2%	Krzysztof Kozlowski	191	1.3%	Hawking Zhang	68106	8.4%
Intel	1540	10.8%	Qualcomm	73226	9.0%	Ian Rogers	133	0.9%	Kees Cook	22257	2.7%
Google	1075	7.5%	(Unknown)	70208	8.6%	Christoph Hellwig	126	0.9%	Vikas Gupta	19074	2.3%
AMD	943	6.6%	Google	68169	8.4%	Eric Dumazet	108	0.8%	Ethan Nelson-Moore	17685	2.2%
Red Hat	922	6.5%	Intel	50468	6.2%	Andy Shevchenko	103	0.7%	Linus Torvalds	15884	1.9%
Qualcomm	795	5.6%	Red Hat	40147	4.9%	Jani Nikula	102	0.7%	Taniya Das	14182	1.7%
(None)	598	4.2%	Broadcom	24765	3.0%	Rafael J. Wysocki	96	0.7%	Likun Gao	14151	1.7%
Meta	424	3.0%	(None)	23184	2.8%	Lijo Lazar	92	0.6%	Alex Deucher	12744	1.6%
SUSE	362	2.5%	Meta	20655	2.5%	Eric Biggers	90	0.6%	Eric Biggers	11657	1.4%
Oracle	296	2.1%	IBM	18792	2.3%	Thomas Weißschuh	88	0.6%	David Howells	11341	1.4%
Huawei Technologies	291	2.0%	Oracle	17101	2.1%	Rob Herring	86	0.6%	Claudio Imbrenda	10514	1.3%
(Consultant)	276	1.9%	Linux Foundation	17084	2.1%	Uwe Kleine-König	84	0.6%	Pavankumar Nandeshwar	9631	1.2%
NVIDIA	268	1.9%	NXP Semiconductors	15380	1.9%	Sean Christopherson	81	0.6%	Ian Rogers	7967	1.0%
Renesas Electronics	258	1.8%	Collabora	14691	1.8%	Thorsten Blum	77	0.5%	Vladimir Zapolskiy	7734	0.9%
IBM	256	1.8%	SUSE	13113	1.6%	Filipe Manana	74	0.5%	Detlev Casanova	6508	0.8%
Linaro	245	1.7%	Linaro	12100	1.5%	Al Viro	68	0.5%	Lijo Lazar	5777	0.7%
NXP Semiconductors	238	1.7%	Huawei Technologies	10526	1.3%	Jakub Kicinski	67	0.5%	Rob Herring	5670	0.7%
Collabora	226	1.6%	NVIDIA	9954	1.2%	Alice Ryhł	67	0.5%	Harsh Kumar Bijlani	5027	0.6%
Arm	223	1.6%	Renesas Electronics	8636	1.1%	Randy Dunlap	63	0.4%	Dmitry Baryshkov	4909	0.6%
Bootlin	144	1.0%	Realtek	8415	1.0%	Dmitry Baryshkov	62	0.4%	Pratik Vishwakarma	4644	0.6%

<https://lwn.net/Articles/1066723/>

Kernel development statistics (kernel 2.6.27)

Kernel 2.6.27 (release 2008-10-7) contributors

Most active 2.6.27 employers

By changesets			By lines changed		
(None)	1925	18.2%	IBM	207215	18.1%
Red Hat	1405	13.2%	(None)	129998	11.4%
(Unknown)	921	8.7%	Red Hat	109970	9.6%
IBM	791	7.5%	(Unknown)	108878	9.5%
Intel	605	5.7%	Nokia	52022	4.5%
Novell	586	5.5%	Novell	49944	4.4%
Movial	234	2.2%	(Consultant)	46529	4.1%
SGI	197	1.9%	Broadcom	43438	3.8%
(Consultant)	193	1.8%	Atheros	38212	3.3%
Sun	184	1.7%	Movial	35439	3.1%
XenSource	165	1.6%	Intel	32887	2.9%
Parallels	157	1.5%	Freescale	25511	2.2%
Oracle	148	1.4%	SGI	23444	2.0%
Marvell	143	1.3%	Marvell	20967	1.8%
Fujitsu	138	1.3%	Renesas Technology	15723	1.4%
AMD	129	1.2%	MIPS Technologies	15701	1.4%
Renesas Technology	125	1.2%	Pengutronix	13334	1.2%
linutronix	121	1.1%	Atmel	10786	0.9%
Simtec	119	1.1%	Analog Devices	10725	0.9%
(Academia)	108	1.0%	Sun	9176	0.8%

Most active 2.6.27 developers

By changesets			By changed lines		
Ingo Molnar	238	2.2%	Paul Mackerras	138374	12.1%
Bartlomiej Zolnierkiewicz	235	2.2%	David Woodhouse	44759	3.9%
Adrian Bunk	221	2.1%	Jean-Francois Moine	41157	3.6%
David S. Miller	206	1.9%	Adrian Bunk	35160	3.1%
Alan Cox	196	1.8%	Artem Bityutskiy	34545	3.0%
Yinghai Lu	192	1.8%	Luis R. Rodriguez	31825	2.8%
Jeremy Fitzhardinge	162	1.5%	Sam Ravnborg	27443	2.4%
Tomas Winkler	128	1.2%	Karsten Keil	24674	2.2%
Ben Dooks	120	1.1%	Russell King	22861	2.0%
Jean Delvare	113	1.1%	Eilon Greenstein	19470	1.7%
Steven Rostedt	108	1.0%	Alan Cox	16957	1.5%
Harvey Harrison	105	1.0%	Felipe Balbi	16287	1.4%
Pavel Emelyanov	103	1.0%	Kumar Gala	14490	1.3%
Thomas Gleixner	101	1.0%	David Brownell	12551	1.1%
Jean-Francois Moine	89	0.8%	Ralf Baechle	11057	1.0%
Lennert Buytenhek	88	0.8%	Lennert Buytenhek	9735	0.9%
Hans Verkuil	81	0.8%	David S. Miller	8621	0.8%
Joerg Roedel	81	0.8%	Juergen Beisert	8516	0.7%
Arnd Bergmann	76	0.7%	Steven Rostedt	8455	0.7%
David Brownell	75	0.7%	Ben Dooks	8399	0.7%

<https://lwn.net/Articles/302061/>

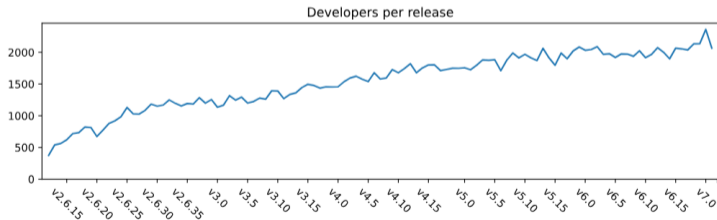
Recent Linux kernel development, "Quite well organized"

Early days of kernel development, "Highly fragmented"

How does community manage, "fragmented to stabilized migration"

Let's go back to the origin to understand the essence of OSS

Version	2.6.27	7.0	diff
Release date	2008.10.7	2026.4.12	17.5 years
Code size (LOC)	9.0M – 10.0M	39.6M	4x
Contributors	1,109	2,362	2.1x
Development Period	88 days	63 days	



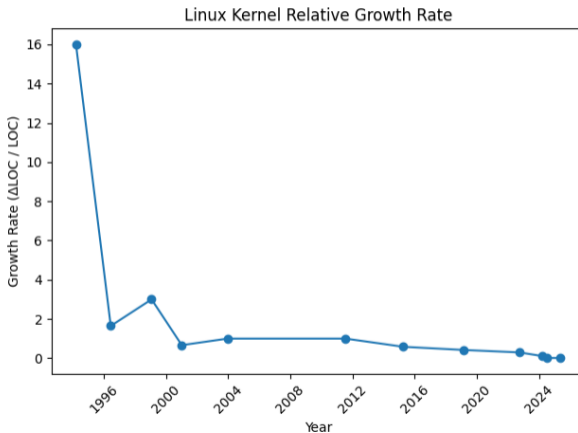
<https://lwn.net/ksdb/releases/>

Recent Linux kernel development, "Quite well organized"

Early days of kernel development, "Highly fragmented"

How does community manage, "fragmented to stabilized migration"

Growth rate

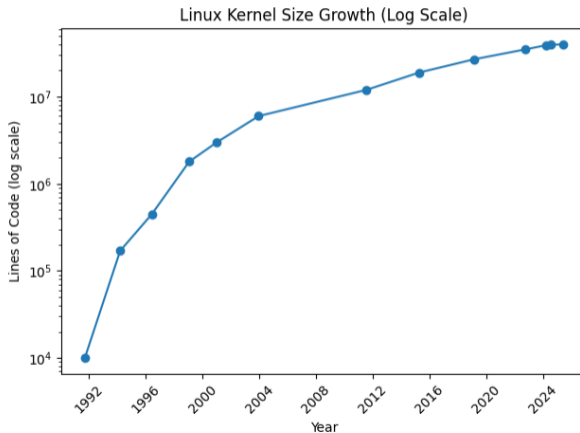


Recent Linux kernel development, "Quite well organized"

Early days of kernel development, "Highly fragmented"

How does community manage, "fragmented to stabilized migration"

Code size stabilization



Conclusion: AGL has potential to truly open the door for SDV

The challenge of SDV is not only software size. It is integration scalability

- Linux solved this problem over 20 years through:
 - distributed ownership
 - staged integration
 - automated validation
 - time-based release engineering
- AGL has the opportunity to bring these lessons into the automotive industry.