SDV 時代の持続可能なソフトウェア開発を支える OSS と Upstream First

Elektrobit SDV Forum Japan 2026

宗像尚郎
hisao.munakata.vt(at)renesas.com

ルネサスエレクトロニクス株式会社ハイパフォーマンスコンピューティングプロダクトグループ
SoC ソフトウエアイネーブルメント部

2026-11-26

自己紹介（講演者の立ち位置）

ルネサスで R-Car 向けの OSS SW 基盤の開発 を行っています

- ルネサスエレクトロニクス株式会社 での活動
  - 最新の自動車向けの SW 要件（Requirement）収集 → ソリューション検討
  - クラウドや AI など SW 技術トレンド理解 → 要素技術開発（仮想化、機能安全等）
  - 社内外 SW 開発部門に対するガイダンス → OSS 活用方法の指南
- オープンソース開発コミュニティ での活動（会社公認の社外活動）
  - AGL (Automotive Grade Linux) プロジェクト、Yocto プロジェクト理事
  - COVESA（Connected Vehicle System Alliance）理事
  - OSVDI（Open SDV Initiative）メンバー
  - SOAFEE、Jaspar、AUTOSAR などの 標準化団体とのコラボレーション推進
  - 各種 OSS プロジェクト、個人開発者との連携（コミュニティ開発リソースの活用）

「産業界における OSS 利活用を多面的にサポート」してきた OSS の支援者です

何故 SoC の Upstream が重要になったのか

Upstream kernel 開発に参加している企業の変遷 （2009 → 2026）

2009.12

Most active 2.6.32 employers

By changesets

(None)	1845	17.1%
Red Hat	1028	9.5%
(Unknown)	933	8.7%
Intel	888	8.2%
Novell	662	6.1%
IBM	603	5.6%
Oracle	319	3.0%
* Renesas Technology	264	2.5%
AMD	251	2.3%
Nokia	204	1.9%
Fujitsu	201	1.9%
Atheros Communications	197	1.8%
(Consultant)	195	1.8%
(Academia)	167	1.6%
* Texas Instruments	155	1.4%
Wolfson Micro	153	1.4%
* Broadcom	149	1.4%
HP	130	1.2%
* Analog Devices	124	1.2%
Pengutronix	119	1.1%

By lines changed

(None)	282017	18.6%
Novell	256808	16.9%
Red Hat	150781	9.9%
* Broadcom	84904	5.6%
Intel	79267	5.2%
(Unknown)	77122	5.1%
Brocade	49189	3.2%
Logic Supply	45165	3.0%
Google	40936	2.7%
IBM	29616	2.0%
Wolfson Micro	25577	1.7%
* Texas Instruments	24824	1.6%
* Renesas Technology	24507	1.6%
Nokia	24192	1.6%
Microsoft	19696	1.3%
Oracle	19410	1.3%
(Consultant)	18774	1.2%
Conexant	16790	1.1%
LinSysSoft Technologies	15173	1.0%
GE Fanuc	12495	0.8%

https://lwn.net/Articles/363456/

2026.10

Most active 7.3 employers

By changesets

(Unknown)	3851	20.5%
AMD	1186	6.3%
Intel	1179	6.3%
Google	1138	6.1%
* Qualcomm	1115	5.9%
Red Hat	837	4.5%
(None)	679	3.6%
* Kylinsoft	624	3.3%
Meta	524	2.8%
IBM	472	2.5%
* NVIDIA	400	2.1%
SUSE	363	1.9%
(Consultant)	303	1.6%
Oracle	271	1.4%
(Academia)	268	1.4%
* Renesas Electronics	240	1.3%
* NXP Semiconductors	237	1.3%
* Arm	234	1.2%
Amutable	216	1.2%
Huawei Technologies	174	0.9%

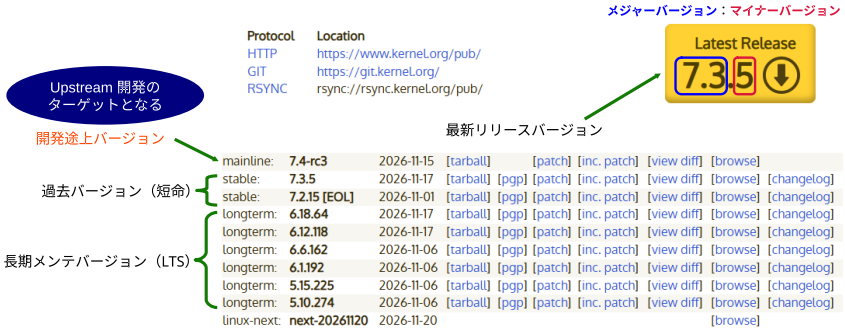
By lines changed

AMD	225173	22.3%
(Unknown)	144387	14.3%
* Qualcomm	61771	6.1%
Google	56977	5.6%
* NXP Semiconductors	45506	4.5%
Intel	44774	4.4%
Red Hat	38420	3.8%
(None)	32075	3.2%
Meta	27768	2.7%
IBM	20441	2.0%
* NVIDIA	16737	1.7%
Realtek	15731	1.6%
Amutable	14873	1.5%
Morse Micro	13949	1.4%
HP Enterprise	13698	1.4%
* Renesas Electronics	11756	1.2%
(Consultant)	11405	1.1%
Cloudflare	10070	1.0%
* Collabora	8405	0.8%
Samsung	8399	0.8%

https://lwn.net/Articles/1100241/

従来より開発を主導してきたサーバー系企業に加え、近年はチップベンダーの開発貢献が顕著に

Linux kernel 総本山 (kernel.org) には何が公開されているのか？



https://kernel.org/

組み込みチップベンダーは 自社チップ向けの Linux BSP を提供する

インハウスで開発されたローカルな BSP

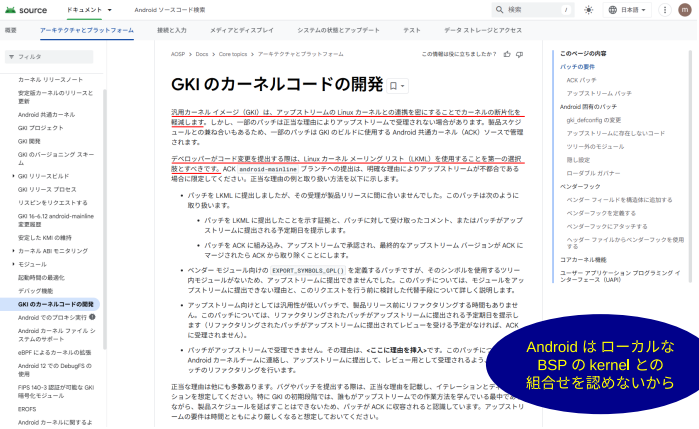
- 公開済みの Upstream kernel を入手
  - コミュニティ開発は終わっている
- fork = Upstream から乖離する
- 自社 SoC 向けのコードを追加
- ローカルに BSP を顧客に提供
  - メインラインの 進化に追従しない
  - 脆弱性パッチ への対応が困難
  - ベンダー毎に フラグメント する

Upstream ベースのオープンな BSP

- コミュニティの開発プロセス に参加
- 開発途上バージョンにパッチ投稿
- コミュニティ内で 公開コードレビュー
- 誰でも複数回のコード修正が必要
  - マージまで半年〜一年 かかる
  - チップリリースからの 時間差がある
- Upstream コードから BSP を作成

組み込みチップベンダーにとって Upstream 開発への参加は極めて大きな負担である

それでも 何故チップベンダーがコミュニティ開発に参加する？ （1）



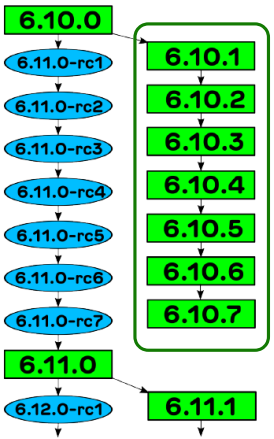
Android は ローカルな BSP の kernel との 組合せを認めないから

https://source.android.com/docs/core/architecture/kernel/kernel-code?hl=ja

それでも 何故チップベンダーがコミュニティ開発に参加する？（2）

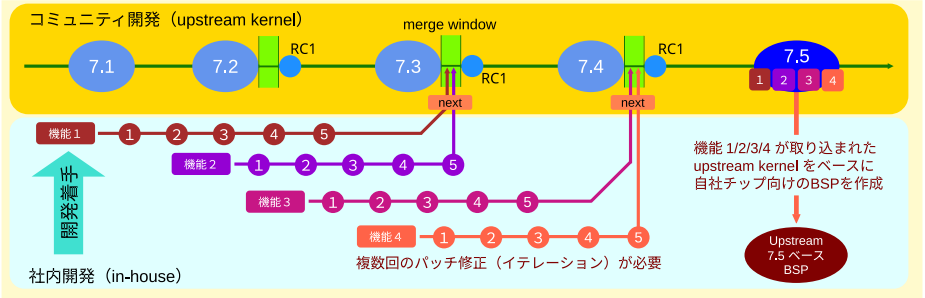
コミュニティがリリースする 脆弱性パッチを活用したい

- コミュニティでは Stable Release をリリースしている（右図の緑線で囲まれた部分、ほぼ毎週更新されている）
- Stable Release には以下が含まれる
  - 重要で確実（副作用がない）バグ修正
  - 脆弱性対策パッチ
- kernel を fork すると stable patch が適用できない が、膨大な脆弱性パッチをローカル開発するのは不可能
- CRA（欧州サイバーレジリエンス法）対応などの目的でコミュニティの脆弱性対策パッチを利用したい
- https://git.sr.ht/~gregkh/presentation-security/tree/b6150729a7a06447b1e8d01a6fb4f8a2be066933/item/security-stuff.pdf



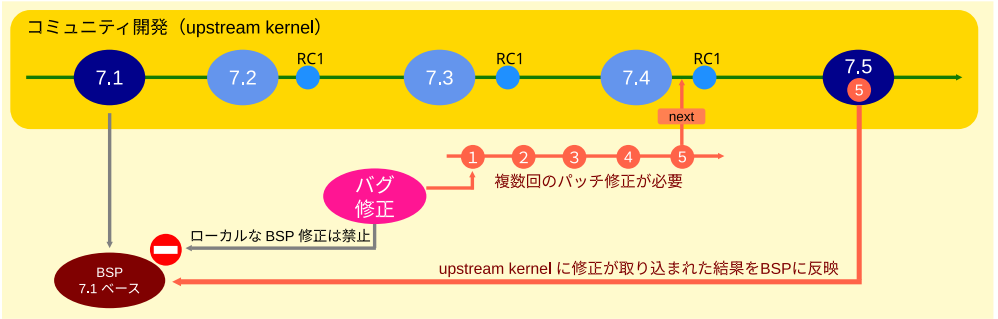
Upstream First という考え方

Upstream First な開発プロセス とは



- 自社チップを動かすために 必要な機能（機能１～４）を kernel に追加するための提案（RFC）をパッチとして開発コミュニティに公開メーリングリスト（LKML）を通じて投稿する。主にデバイスドライバー関係の機能追加
- コミュニティのメンテナーは提案内容を確認し、コーディングの改善や kernel 内の各種フレームワークとの整合性などを確認し、コードレビュー結果を返す。通常このやり取り（イテレーション）は数回～十数回繰り返される。
- レビューが完了したパッチは、各カーネルバージョンの開発初期（＝RC1 がリリースされるまで）の二週間程度のマージウィンドウ期間に取り込まれて、他の追加コードと一緒に検証期間にはいる。通常 RC7 くらいまでいく。
- 最終的に必要な機能がすべて取り込まれたバージョンの kernel を使って BSP を作成し、ユーザーにリリースする。

フラグメントを回避する BSP のバグ修正プロセス（とても時間がかかる）



Upstream First のメリット	Upstream First のデメリット
コミュニティレビューで コード品質が改善 する	時間がかかる（BSP修正に最大半年かかるケースも）
以降のkernel バージョンでは ローカルパッチが不要	公開レビューの対象になる（機密を保持できない）
インハウスコード や ローカルFix が排除される	レビュー期間中に何回もコード修正が必要になる

新規チップ開発 と Upstream First の 不都合な関係性

SW First というけれど....

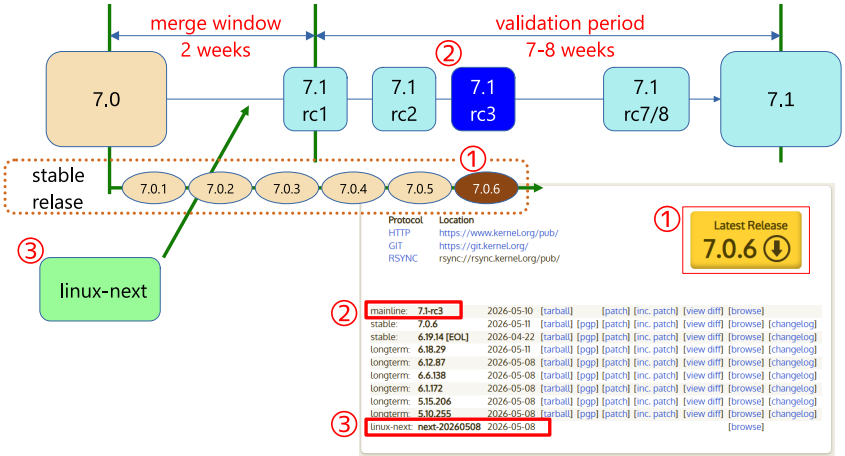
- チップベンダーは チップが完成した時点で SW (BSP) が完成 していてほしい
- 実チップができる前から SW 開発を先行させる 仮想開発環境 も利用できる
- 仮想環境上で Linux kernel 起動 まで確かめている＝チップの検証にも FB できる
  - 仮想環境上の Arm A720 Hunter 32 コアで kernel を起動させた実績あり
- 開発コミュニティは 仮想環境上で動作するコードはレビューしてくれない
- メンテナーはコードを実際に動作させられる 検証環境（評価ボード）を要求する
- 開発途上の 非公開なチップの仕様、名称等を公開 ML に投稿できない ケースも

Upstream First な BSP をチップ完成直後に提供する事は原理的に不可能である

kernel のリリースは原則として日曜日 （なので 63 日 or 70 日周期）

Release	Date	Duration	RCs	Developer	Commits		
					Total	Reg.	Merge
v7.3	2026-10-18	63	7	3,021	20,260	18,765	1,495
v7.2	2026-08-16	63	7	2,652	17,833	16,418	1,415
v7.1	2026-06-14	63	7	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

近年の Upstream kernel 開発は厳密な時間厳守 で進行する



Upstream First のベストプラクティス（良くある勘違い）

LTS バージョンを狙い撃ちした開発 はできない

Active kernel releases

There are several main categories into which kernel releases may fall:

2026-02-25
In Releases.

Prepatch
Prepatch or "RC" kernels are mainline kernel pre-releases that are mostly aimed at other kernel developers and Linux enthusiasts. They must be compiled from source and usually contain new features that must be tested before they can be put into a stable release. Prepatch kernels are maintained and released by Linus Torvalds.

Mainline
Mainline tree is maintained by Linus Torvalds. It's the tree where all new features are introduced and where all the exciting new development happens. New mainline kernels are released every 9-10 weeks.

Stable
After each mainline kernel is released, it is considered "stable." Any bug fixes for a stable kernel are backported from the mainline tree and applied by a designated stable kernel maintainer. There are usually only a few bugfix kernel releases until next mainline kernel becomes available -- unless it is designated a "longterm maintenance kernel." Stable kernel updates are released on as-needed basis, usually once a week.

Longterm
There are usually several "longterm maintenance" kernel releases provided for the purposes of backporting bugfixes for older kernel trees. Only important bugfixes are applied to such kernels and they don't usually see very frequent releases, especially for older trees.

Longterm release kernels

Version	Maintainer	Released	Projected EOL
6.18	Greg Kroah-Hartman & Sasha Levin	2025-11-30	Dec, 2028
6.12	Greg Kroah-Hartman & Sasha Levin	2024-11-17	Dec, 2028
6.6	Greg Kroah-Hartman & Sasha Levin	2023-10-29	Dec, 2027
6.1	Greg Kroah-Hartman & Sasha Levin	2022-12-11	Dec, 2027
5.15	Greg Kroah-Hartman & Sasha Levin	2021-10-31	Dec, 2026
5.10	Greg Kroah-Hartman & Sasha Levin	2020-12-13	Dec, 2026

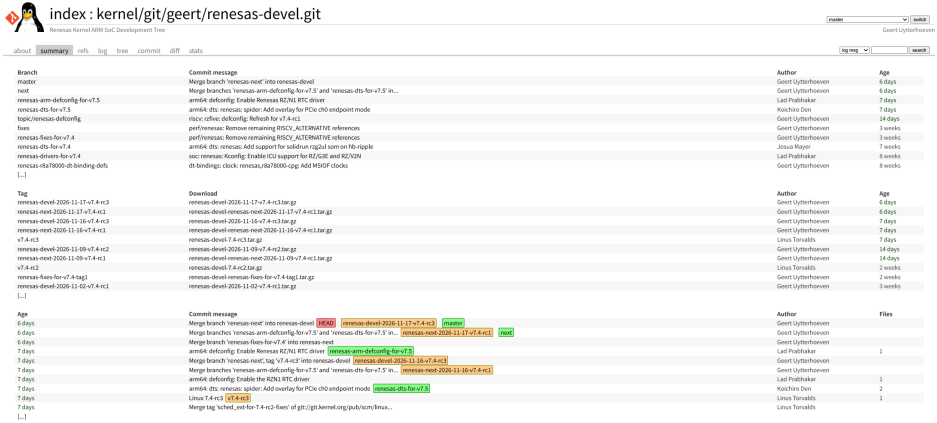
https://www.kernel.org/category/releases.html

LongTerm Support (LTS) バージョンとは

- 通常バージョン のメンテ期間は２つ先のバージョンまで（19〜20 週）
- 毎年１つ長期メンテ版（longterm）を決め、（今は）３年間 メンテを維持する
- どのバージョンが longterm になるかは 事前には知らされない
- 先にわかると、そのバージョンにだけパッチを送る人が出てくるから

すべてのバージョンに対して途切れなく開発を継続することが重要である

すべてのバージョンを対象とした Upstream 開発 （ルネサスの実例）



https://git.kernel.org/pub/scm/linux/kernel/git/geert/renesas-devel.git/

アップストリームすれば あとは放置で良いか？

負担が軽減される 作業

- 新しい kernel への追従
- kernel 共通部分の性能検証等
- ローカルパッチの維持・管理
- セキュリティ修正のバックポート
- BSP バージョン移植コスト削減
- BSP (kernel) 更新内容の個別発信

継続した対応が必要 な作業

- kernel API 変更への対応
- kernel フレームワーク変更への対応
- 新しい kernel での動作検証
- リグレッションの調査・修正
- メンテナーの検証活動に対する支援
- 開発コミュニティとの継続連携

コードがマージされたら、コミュニティと共にそれらを保守していく必要がある

ところで、2026 年を振り返る

ポジティブ影響

- 一億総プログラマー時代へ
  コーディング 経験が無くてもOK
- **AI エージェント** の活用
  複雑なプロセスの自動化
- 会議 **議事録** 作成、**翻訳** 等
  会社の仕事を大きく補佐
- **フィジカルAI** への応用
  自律型の物理機械制御

２０２６年
AI 民主化元年
日常生活にAIが実装された

懸念事項

- **サイバーセキュリティ**事案
  AI を使った自動アタック
- **AI 暴走**の懸念
  説明性欠如と監視機能
- **個人情報の流失**
  未知のセキュリティホールの発見
- 膨大な **電力消費**
  CPU、GPU、Memory **逼迫**

AI が理解できる”オープンなコード、データ、ドキュメントの整備”を考える必要がある

まとめ

- これまで Linux kernel 開発を担っていたのはサーバー／ IT 系企業のエンジニアであったが、チップベンダーの貢献度が非常に高まってきた 背景を考察した.
- チップベンダーが Upstream First ポリシーに基づいた BSP 開発を行うのは容易ではない が、近年のサイバーセキュリティ対応（CRA など）を考えると 真面目に取り組まなければならないところまできている と認識する必要がある.
- そして、AI 全盛時代においては AI が読めるコード、データ、ドキュメントの整備が非常に重要 になっている. このような観点からも Upstream First への取り組みが別の新しい意味での重要性をもってきた事に気づく必要もあるだろう.

Upstream First は OSS 開発手法に留まらず、ソフトウェアのフラグメンテーション解消、長期メンテナンス性とセキュリティを確保する持続可能な開発モデルである