そのオープンソース、製品開発に使って大丈夫ですか？

Automotive World 2025：SDV Forum-X 講演 [SDV-9]

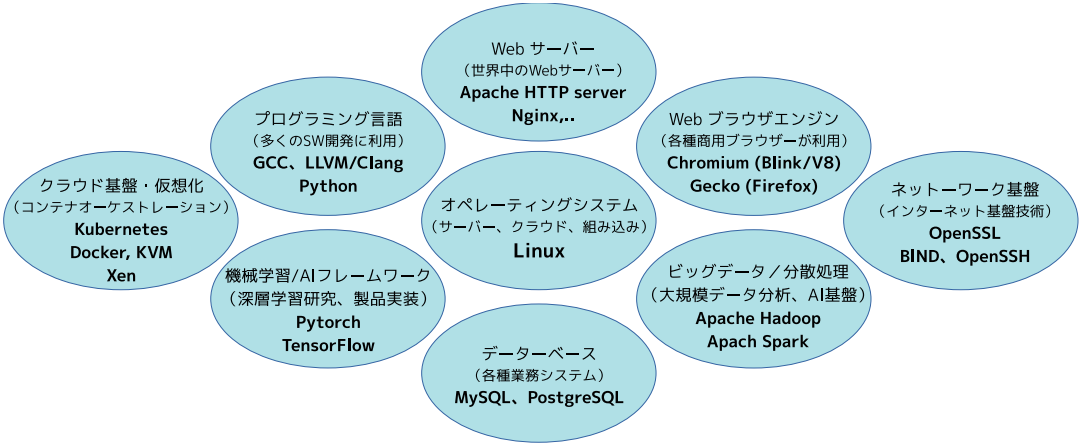
宗像尚郎
hisao.munakata.vt(at)renesas.com

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　シニアダイレクタ

2025-9-19

OSS の基本を再確認しよう（導入編）

私達の日常はもはや「OSS 無しでは成り立たなくなっています」



OSS について聞かれたら、自信を持って説明できますか？

「特別な契約締結も求められずに無償で使える SW」程度に考えていませんか？

- 例えば、以下のような「不適切な認識」を耳にすることが良くあります
- ［ライセンス］OSS を製品開発に利用する時には契約を気にする必要はない
- ［費用］OSS を使えば費用がかからない
- ［品質］OSS は商用 SW より品質が悪い
- ［保証］商用 SW には保証があるが、OSS は無保証である
- ［保守］OSS にコードを公開すれば、後はコミュニティが保守してくれる
- 導入編では、まずこれらの誤解について一つずつ解消していきます

自動車業界にはまだ「OSS の正しい理解が充分波及していない」印象があります

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

”違和感” を感じますか？　（自動車業界だけの問題でもないようです）

デジタル・ガバメント推進標準ガイドライン 実践ガイドブック
2025 年（令和 7 年）5 月 27 日 デジタル庁

オープンソースソフトウェアの特徴を理解して採用する

オープンソースソフトウェア（OSS）には、先進的な機能が利用できるメリットがある一方で、不具合があってもサポートを受けられないなどのデメリットもあります。メリットとデメリットの両方を正しく理解した上で、プロジェクトの特性に合わせて、OSS の採用を検討しましょう

メリット		デメリット	
拡張性	・ 公開されているソースコードをもとに、不具合の修正や機能拡張などを行うことができる。	コンプライアンス	・ OSS を利用して独自に開発したアプリケーションについてもソースコードを開示する義務が生じる可能性がある。 ・ OSS 開発者へ損害賠償請求等ができない。 ・ ライセンス違反を理由に第三者から訴訟を起こされる可能性がある。
コスト（※）	・ ライセンス料がかからず、導入コストを抑えられる。 ・ ベースとなる機能や部品として利用することで、開発工数を削減できる。		
先進性	・ 先進的な機能が利用できることも多い。	サポート	・ 緊急時のサポートを受けられない。
セキュリティ	・ 市販のソフトウェア等では、ソースコードを確認することができないが、OSS では、ソースコードが公開されており、脆弱性等を直接確認することができる。	セキュリティ	・ ソースコードが公開されているため、脆弱性を突いた攻撃を受ける可能性がある。
品質	・ 多くのユーザが利用しており、活動が活発な OSS の場合は安定した品質を期待できる。	不具合修正	・ 活動が停滞している OSS の場合、不具合対応されない場合がある。

OSS ではあるものの、製品自体が有償化されていたり、OSS の入手は無償であってもサポートなどが有償化されていたりする場合があるため、OSS の採用を検討する際にコストを確認することが重要です。また、以下の理由で、管理コストが割高になる可能性があることに注意が必要です。
※ OSS はサポート期間が一般的に短いものが多いため、バージョンアップなどの対応が増える場合があります。

https://www.digital.go.jp/assets/contents/node/basic_page/field_ref_resources/e2a06143-ed29-4f1d-9c31-0f06fca67afc/ae9

OSSの “ライセンス” に関わる誤解を解く

”OSS を製品開発に利用する時には契約は関わらない”

［ライセンス］再配布する時にはライセンスに注意する必要があります

「オープンソース配布ライセンス」の内容を正しく理解する 必要があります

- ソースコードには、必ず一緒に ライセンスファイル が提供されています
- OSS の 内部利用には契約は不要（多くのライセンスで自由な利用が認められます）
- OSS を 再配布する時に OSS ライセンスの規定を遵守 する必要があります
- OSS ライセンスは一種類ではなく、ライセンス条件の詳細はそれぞれで異なります
- 以下のような行為は OSS ライセンス違反になります
  - OSS コードを参照して商用 SW 開発して配布する
  - OSS コードのライセンスを変更または削除して配布する
  - OSS コード部分だけを抽出したドキュメントを配布する

自社製品の制御に OSS を利用した場合は OSS の「再配布」に該当します

［ライセンス］コンピュータ プログラムは著作物 に分類されます

著作権／ Copyright (著作財産権、著作者人格権) の適用対象となるもの（例）

- 言語 (論文、小説、脚本、詩歌、俳句、講演など)
- 音楽 (楽曲及び楽曲を伴う歌詞)
- 美術 (絵画、版画、彫刻、漫画、書、舞台装置など）
- 建築（芸術的な建造物（設計図は図形の著作物）
- 地図や図形 (地図と学術的な図面、図表、模型など）
- 映画 (劇場用／テレビ、ビデオソフト、ゲームソフト)
- 写真、グラビアなど
- コンピュータ プログラム

コンピュータ プログラムの自由な再利用には「著作権による制約」がかかります

［ライセンス］「著作財産権」(＝ 狭義の著作権) は対価の請求権 です

著作権（財産権）

複製権 （第21条）	著作物を印刷、写真、複写、録音、録画などの方法によって有形的に再製する権利
上演権・演奏権 （第22条）	著作物を公に上演したり、演奏したり（録音物や録画物を再生することを含む）、また、それらの上演、演奏された著作物を電気通信設備を用いて公に伝達する権利
上映権 （第22条の2）	著作物を公にスクリーンやディスプレイに映写する権利
公衆送信権・公の伝達権 （第23条）	著作物を自動公衆送信(*)したり、放送したり、有線放送したり、また、それらの公衆送信された著作物を受信装置を用いて公に伝達する権利 ＊自動公衆送信とは、サーバなどに蓄積された情報を公衆からのアクセスに応じ自動的に送信することをいいます。また、そのサーバに蓄積された段階を送信可能化といいます。
口述権 （第24条）	言語の著作物を朗読などの方法により口頭で公に伝える（口述の録音物や録画物を再生することを含む）権利
展示権 （第25条）	美術の著作物と未発行の写真の著作物の原作品を公に展示する権利
頒布権 （第26条）	映画の著作物の複製物を頒布（販売・貸与など）する権利
譲渡権 （第26条の2）	映画以外の著作物の原作品又は複製物を公衆へ譲渡する権利 （ただし、いったん適法に譲渡された場合は、その後の譲渡には譲渡権は及びません）。
貸与権 （第26条の3）	映画以外の著作物の複製物を公衆へ貸与する権利
翻訳権・翻案権など （第27条）	自己の著作物を翻訳、編曲、変形、翻案等する権利（二次的著作物を創作する権利）
二次的著作物の利用権 （第28条）	自己の著作物を原作品とする二次的著作物を利用（上記の各権利に係る行為）することについて、二次的著作物の著作者が持つものと同じ権利

公的社団法人著作権情報センター「著作者にはどんな権利がある？」(2025/09/04)

https://www.cric.or.jp/qa/hajime/hajime2.html

［ライセンス］「著作者人格権」は著作者の心情を保護する権利です

著作者人格権は 第三者によるコードの改変や再配布を制約 しています

- 公表権（著作権法 18 条 1 項）
  - 無断で公表されない権利、すなわち未だ公表されていない自分の著作物について、公表するかどうか、いつ、どういう方法及び条件で公表するかを決定する権利です
- 氏名表示権（著作権法 19 条 1 項）
  - 自分の著作物を公表する際に、著作者名を表示するかどうか、どのように表示するか（実名で表示するのか、ペンネームなどの変名で表示するのか）を決定できる権利です
- 同一性保持権（著作権法 20 条 1 項）
  - 自分の 著作物の内容、題号を著作者の意に反して無断で改変させない権利 です

著作者人格権は財産権とは異なり、第三者に譲渡したり相続することはできません

［ライセンス］オープンソースの起点となった二つの重要な出来事

GNU の”自由ソフトウエア運動”（1985）

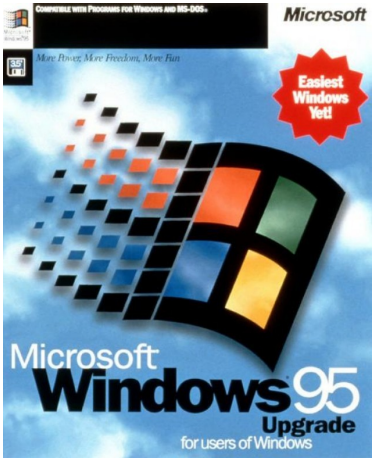
- Richard Stallman が牽引
- GNU 宣言
- ハッカーを擁護（SW の自由な研究）
- SW の利用目的、解析、改造、再配布、研究の自由が担保されるべき と主張
- Copyleft の概念 を定義
- GPL/LGPL ライセンス
- 改変したソースコードの開示義務

Netscape Browser ソース公開（1997）

- Netscape 社の戦略判断
- Microsoft IE への対抗措置
- 大規模商用 SW のソース開示
- NPL/MPL ライセンス
- 新規追加ファイルには公開義務なし
- 後に OSI（Open Source Initiative）が NPL/MPL を OSS ライセンスと承認
- OSI が OSS というブランド を定義

この２つの重要なイベントのおかげで「今のオープンソースの隆盛」があるのです

［ライセンス］Internet Explorer vs. Netscape Navigator 対決



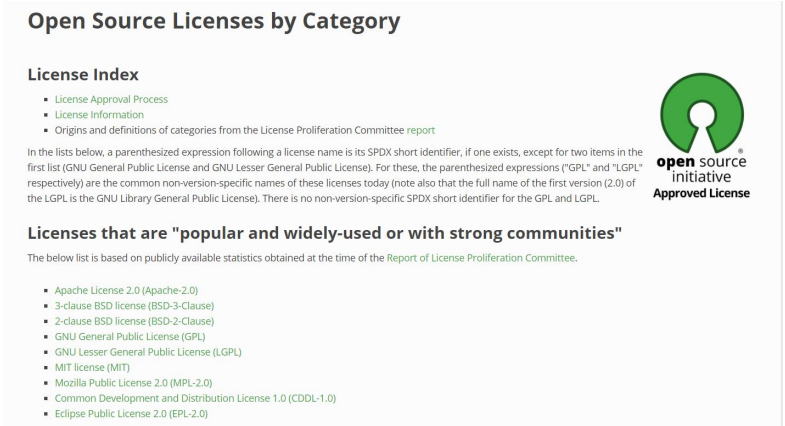
［ライセンス］オープンソースの定義 (v1.9) 注釈付（八田真行訳）

OSI が OSD (Open Source Definition = オープンソースの定義) を規定しました

- 1. 再頒布の自由
- 2. ソースコード (を入手できること)
- 3. 派生ソフトウェア（を作れて、同じライセンスを付与すること）
- 4. 作者のソースコードの完全性 (integrity)（が担保されること）
- 5. 個人やグループに対する差別の禁止
- 6. 利用する分野 (fields of endeavor) に対する差別の禁止
- 7. ライセンスの分配 (distribution)（に際し追加ライセンスが不要）
- 8. 特定製品でのみ有効なライセンスの禁止
- 9. 他のソフトウエアを制限するライセンスの禁止
- 10. ライセンスは技術中立的でなければならない
- http://www.opensource.jp/osd/osd-japanese.html

2025 年 9 月時点で OSD 互換 OSS ライセンスは 122 種類もあります

［ライセンス］OSI が認定した オープンソースライセンスの例



Open Source Licenses by Category

License Index

- License Approval Process
- License Information
- Origins and definitions of categories from the License Proliferation Committee report

In the lists below, a parenthesized expression following a license name is its SPDX short identifier, if one exists, except for two items in the first list (GNU General Public License and GNU Lesser General Public License). For these, the parenthesized expressions ("GPL" and "LGPL" respectively) are the common non-version-specific names of these licenses today (note also that the full name of the first version (2.0) of the LGPL is the GNU Library General Public License). There is no non-version-specific SPDX short identifier for the GPL and LGPL.

open source initiative
Approved License

Licenses that are "popular and widely-used or with strong communities"

The below list is based on publicly available statistics obtained at the time of the Report of License Proliferation Committee.

- Apache License 2.0 (Apache-2.0)
- 3-clause BSD license (BSD-3-Clause)
- 2-clause BSD license (BSD-2-Clause)
- GNU General Public License (GPL)
- GNU Lesser General Public License (LGPL)
- MIT license (MIT)
- Mozilla Public License 2.0 (MPL-2.0)
- Common Development and Distribution License 1.0 (CDDL-1.0)
- Eclipse Public License 2.0 (EPL-2.0)

https://opensource.org/licenses/category

［ライセンス］コードの開示義務 について ２つの類型 があります

コピーレフト型ライセンス

- 代表は GNU GPL/LGPL
- （共通）改変、再配布、利用を許諾する
- ソースコードの開示義務あり
- Linux kernel など
- OSS コミュニティの活性化に寄与
- GPL は他のライセンスと結合できない

パーミッシブ型ライセンス

- 代表は MIT、Apache2
- （共通）改変、再配布、利用を許諾する
- 改変したコードの開示を義務づけない
- Android、AI 関連ツールなど
- OSS の商用製品への適用を促進
- 一方で分岐（fork）を生み出しやすい

産業界で使いやすい「パーミッッシブ型ライセンス」が好まれる傾向があります

［ライセンス］参考：OSI がオープンだと認めなかったケース

OSS ライセンス提案の否認例

- JSON License
  - The Software shall be used for Good, not Evil
  - 「善用義務」条項は利用制限に該当
- CC-BY-NC
  - NC は明示的に 商用利用を制限
  - 利用制限に該当
- Mulan PSL v1
  - 中国の法律に従う義務を規定
  - 地域依存的な制限に該当

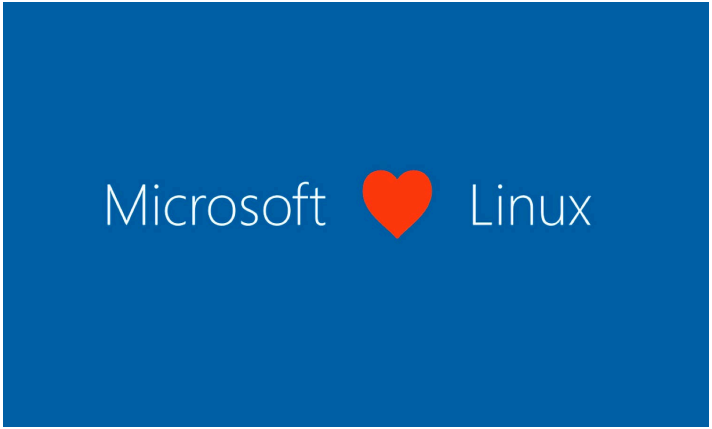
Open Source AI Definition の否認例

- オープンな AI モデルとして紹介されているものでも...
- GPT-2 (OpenAI, 2019)
  - コード、ウェイトを公開済
  - 訓練データの詳細は非公開
  - OSAID 部分適合 の扱い
- LLaMA2 (Meta, 2023)
  - コード、ウェイトを公開済
  - 訓練データは公開＋ Meta 独自取集
  - OSAID 部分適合 の扱い

［ライセンス］以前 Microsoft が推進した Get The Facts キャンペーン

https://gigazine.net/news/20070520_anti_linux_propaganda/

［ライセンス］Microsoft Loves Linux (by CEO Satya Nadella, 2015)



https://www.microsoft.com/ja-jp/cloud-platform/Windows-Server-blog-Loves-Linux.aspx

OSSの “費用” に関わる誤解を解く

”OSS を使えば費用がかからない”

［費用］「OSS なら費用ゼロ」ではない が 総費用削減 は期待できます

無償部分／有償部分の分類

- OSS ライセンスが 無償 を担保しているもの
  - ソースコードの入手
  - ビルド手順情報やビルドに必要なファイルの入手
  - オープンソースの 利用に伴うライセンス費用が発生しない こと
- OSS ライセンスに含まれていないもの＝自己解決が必要となります
  - OSS 導入に関わる技術サポート
  - OSS 利用に関わる Q&A 対応（但し、コミュニティの ML を使った質問は可能です）
  - 障害解析、と問題解決のための SW 改修費用
  - サイバーセキュリティ対策などの メンテナンス費用
  - エンジニアの 教育サポート（但し、無償の優良教育コンテンツがあります）

「ソースが無償」という理由だけで、ここまで OSS が普及したわけはありません！

［費用］OSS の「経済価値は莫大」です（Linux kernel 6.12 の例）

項目	集計結果
ソースコード行数（コメント行を除く）	26,422,792 行
ver 6.11 → ver 6.12 の差分（増分）	149,730 行
ファイル数（ディレクトリーを除く）	87,162 個
sloccount によるソフトウエア価値の試算	$ 1,187,734,209（1兆7,840億円）
ver 6.12 の開発期間	63 日（9 週間）
ver 6.12 の開発に参加した開発者	2,074 人
今回はじめて開発に参加した開発者	335 人
追加された機能（= change set）	13,344 個
ver 6.12 に追加された価値 (sloccount 差分)	$ 7,066,056（10.6 億円）

https://lwn.net/Articles/997959/

［費用］OSS 開発プロジェクトの スポンサーは誰なのか？

莫大な価値を生み出すための投資をするには、資金面の裏付けが必要になります

- 各社が手弁当で開発者を送り込んでいる「草の根プロジェクト」 では
  - 基本的に 個人単位の開発者の貢献 の集約しています
  - 企業の所属する開発者は、業務として OSS 開発プロジェクトに参加 しています
  - 特定企業の事業方針の影響を受けにくく、継続性は比較的担保 されます
  - 開発インフラや運営管理などの 固定費には多くの予算を割り当てられません
- 主に企業からのまとまった資金援助で運営される「企業プロジェクト」 では
  - 特定企業や産業コンソーシアム がまとまった資金を提供しています
  - 元々特定企業に帰属していたプロジェクトをオープン化しているケースもあります
  - 資金面に余裕 があり、積極的にマーケティング活動 を進めるケースも多いのです
  - 企業の 戦略変更 や 合従連衡の影響 を受けやすい面もあります

現代の OSS 開発プロジェクトには「かなり大規模な経済活動」という側面もあります

［費用］大部分の OSS は 企業のエンジニアが開発 しています

Linux kernel 開発に参加している企業

- kernel コードに ”採用された” パッチの数
- 2005 年 4 月～2025 年 7 月（20 年超）の累計値
- パッチ数の合計＝ 1,265,698 件
- 参加企業の合計＝ 907 社

- 当初はサーバー系企業が牽引していました
- 最近はデバイスメーカの貢献が顕著になっています
- ルネサスでは 10～20 名の専従開発チーム で OSS の Upstream 活動を進めています

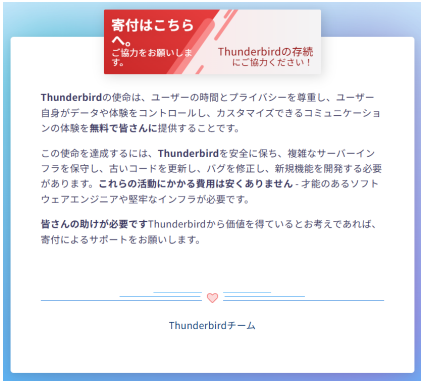
No.1	Unknown	239,127	18.89%
No.2	Intel	132,016	10.43%
No.3	Red Hat	97,489	7.70%
No.4	Hobbyists	79,729	6.30%
No.5	Novell	48,974	3.87%
No.6	Linaro	46,423	3.67%
No.7	IBM	40,870	3.23%
No.8	AMD	40,483	3.20%
No.9	Google	37,618	2.97%
No.10	Huawei	26,651	2.11%
No.11	Renesas Electronics	24,606	1.94%
No.12	Oracle	22,952	1.81%
No.13	Texas Instruments	18,634	1.47%
No.14	Samsung	18,585	1.47%
No.15	NVIDIA	15,597	1.23%
No.16	ARM	12,713	1.00%
No.17	NXP	12,399	0.98%
No.19	Mellanox Technologies	11,521	0.91%
No.21	QUALCOMM	10,027	0.79%
No.23	Broadcom	9,654	0.76%
No.27	Linux Foundation	7,718	0.61%
No.28	Code Aurora Forum	7,079	0.56%
No.29	Analog Devices	6,742	0.53%
No.31	MediaTek	5,647	0.45%
No.33	Marvell	5,275	0.42%
No.35	Freescale	4,699	0.37%
No.38	STMicroelectronics	4,269	0.34%
No.39	Facebook	4,266	0.34%

https://www.remword.com/kps_result/all_whole.html

［費用］OSS 開発プロジェクトの「収支勘定」はどうなっているのか？

プロジェクトの運営には相当な費用がかかる

- 支出
  - IT インフラ費用（各種サーバー運用コスト）
  - Web ンツの開発やメンテ費用
  - メンテナーなど主要専従開発者の給与
- 収入
  - 個人からの寄付
  - スポンサー企業による投資
  - 産業コンソーシアムからの投資
  - 開発者会議などの イベント収入
  - 行政の支援（国プロ、欧州委員会など）



現代の OSS 開発プロジェクトは「開発者の無償貢献だけでは成り立たない」のです

［費用］OSS 開発プロジェクト開発現場の「指揮命令系統」

多くの開発者が参加する OSS 開発プロジェクトの ガバナンス維持は大仕事です

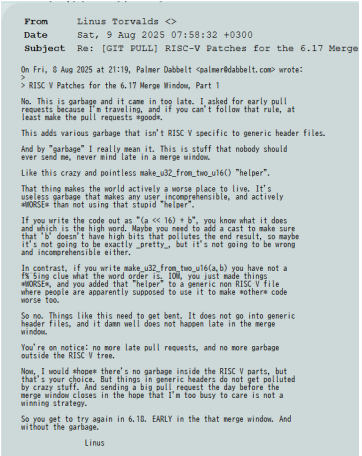
- 世界中の地域からの数千人規模の開発者 が独立に開発を行っています
- 品質レベルや中立性維持 のために「階層化された分業体制」をとっています
- ベロッパー：誰でも投稿可、身元保証や契約が必要な場合もあります
  - 身元保証（DCO = Developer Certificate of Origin）
  - 契約（CLA = Contributor License Agreement）
- レビューワー：投稿内容を 開発者同士でクロスチェック して改善を促します
- サブシステムメンテナー：各技術領域毎の専門家がコードの採否を決定 します
- マスターメンテナー：サブシステムメンテナーが レビューしたコードを統合 し リリース内容を確定させリリースを宣言 します

OSS では「開発ロードマップやリリース計画が公開されていない」ケースも多いのです

［費用］メンテナーは プロジェクトの完全性を厳格に管理 しています

kernel 6.17 開発中に RISC-V 関連パッチ受け取りを拒否

- #define make_u32_from_two_u16(a, b) というヘルパー関数の追加をリクエストされ、これに対して
  - pull_request（マージ要求）のタイミングが遅すぎる
  - 何をする関数かわかりにくい、普通に ((a << 16) + b) と書いた方がずっと可読性がよい（望ましい）
  - このヘルパー関数を RISC-V 以外も利用する共通部に配置するのはさらに良くない
- メンテナーは “Garbage（ゴミ）” と切り捨てました
- あるべき記述法に書き直した上で 6.18 開発時に時間を守って再投稿するよう促しています
- https://lkml.org/lkml/2025/8/9/76



From Linus Torvalds <>
Date Sat, 9 Aug 2025 07:58:32 +0300
Subject Re: [GIT PULL] RISC-V Patches for the 6.17 Merge

On Fri, 8 Aug 2025 at 21:19, Palmer Dabbelt <palmer@dabbelt.com> wrote:
>
> RISC V Patches for the 6.17 Merge Window, Part 1

No. This is garbage and it came in too late. I asked for early pull requests because I'm traveling, and if you can't follow that rule, at least make the pull requests *good*.

This adds various garbage that isn't RISC V specific to generic header files.

And by "garbage" I really mean it. This is stuff that nobody should ever send me, never mind late in a merge window.

Like this crazy and pointless make_u32_from_two_u16() "helper".

That thing makes the world actively a worse place to live. It's useless garbage that makes any user incomprehensible, and actively *WORSE* than not using that stupid "helper".

If you write the code out as "(a << 16) + b", you know what it does and which is the high word. Maybe you need to add a cast to make sure that 'b' doesn't have high bits that pollutes the end result, so maybe it's not going to be exactly _pretty_, but it's not going to be wrong and incomprehensible either.

In contrast, if you write make_u32_from_two_u16(a,b) you have not a f% 5ing clue what the word order is, IOW, you just made things *WORSE*, and you added that "helper" to a generic non RISC V file where people are apparently supposed to use it to make *other* code worse too.

So no. Things like this need to get bent. It does not go into generic header files, and it damn well does not happen late in the merge window.

You're on notice: no more late pull requests, and no more garbage outside the RISC V tree.

Now, I would *hope* there's no garbage inside the RISC V parts, but that's your choice. But things in generic headers do not get polluted by crazy stuff. And sending a big pull request the day before the merge window closes in the hope that I'm too busy to care is not a winning strategy.

So you get to try again in 6.18. EARLY in the that merge window. And without the garbage.

Linus

［費用］OSS 開発プロジェクトにも 栄枯衰勢 があります

創世記	• 斬新なアイディアを少人数の関係者で共有
黎明記	• OSS開発プロジェクトを形成、ML や github を開設 • リードエンジニアが開発をスタート
発展記	• 多くの関係者の関心を引き、活動がモメンタムを得る • 有能な開発者が多数集まり、コミュニティが拡大
安定記	• 当初の開発目標の大部分が実装され実用運用がスタート • エンスージアスト達はこの段階でプロジェクトを離れ始める
衰退記	• 開発者の減少に伴い、SW更新は脆弱性対応などに絞られる • メンテナー個人に負荷が集中し、バーンナウトなどの問題も
事実上消滅	• MLのトラフィックもSPAMで埋められ、事実上活動は停止 • 他のプロジェクトへの移管などを除き、正式な終了は告知されない

OSSの “品質” に関わる誤解を解く

”OSS は商用 SW よりは品質が悪い”

［品質］OSS は 製品に利用可能な品質レベル なのでしょうか？

既に OSS は「重要な社会インフラの中核」を担っています

- 初期 の OSS は、「愛好家による草の根開発（個人ベース）」 が主流でした
- 最近 は「管理された大規模 SW 開発プロジェクト（企業主導）」 に移行しています
- OSS の品質の拠り所は「コードの透明性」と「完全なトレーサビリティ」です
- 数千人規模の同時開発 支える git という仕組みが使われています
- git は Linux 創始者 Linus が開発したもので、OSS 開発の基盤インフラ です
  - 誰が、何時、何の目的で、どのような変更をしたか が記録されます
  - その記録は 永年保存 されるので、何時でも遡って参照できます
  - 障害発生時には、どの変更が影響しているかを 自動的に追跡して確認 できます
- 機能安全 など一部の産業規格には まだ対応が出来ていません

「OSS の品質が商用 SW より劣っている」という認識は時代錯誤だと思います

［品質］ミッションクリティカル領域で OSS の採用が拡大しています

既に 社会インフラを支える「重要な基盤技術」となっています

- 各種クラウドサービス基盤（Google、Amazon、Facebook、Meta、...）
- 証券・金融（データーベース、ブロックチェイン技術）
- 5G/6G 通信インフラ（SW デファインドネットワーク）
- 産業用ネットワーク通信（OPC UA open62541）
- 医療向け画像処理（https://www.orthanc-server.com/）
- 米国国防省（https://dodcio.defense.gov/open-source-software-faq/ ）
- NASA（https://code.nasa.gov/ ）
- エネルギーインフラ（Colonial Pipeline、本件については後述します）

これに伴って OSS に対する「品質要求」は以前よりもはるかに厳格になっています

［品質］用途のダイバーシティ（多様性）も品質に貢献 しています

「同じ基本ソフトウエア」が IOT 機器からスーパーコンピュータで共有されています

- 「OS に求められる機能」は異なるが、同じ kernel コードが利用されています
  - スーパーコンピュータ ⇒ スケーラビリティ
  - サーバー・データーセンター ⇒ 高信頼
  - スマートフォン ⇒ 省電力、情報セキュリティ
  - テレビ・メディアプレーヤ ⇒ 各種メディアフォーマットに対応
  - 車載マルチメディア・ナビ ⇒ 画像と音声を別々に制御
  - ロボット制御 ⇒ リアルタイムメカトロ制御
  - IoT 機器 ⇒ ネットワーク、ファイルシステム
- ストレージや RAM のコスト削減により Linux の適用範囲が拡大 しています
- 結果として 機器間の接続性の確保や、業界標準へのレファレンス対応 が容易に

さまざまな異なる要求が一つのコードに集約されたことにより品質が向上しました

［品質］OSS は リアルタイム制御 にも使えるのでしょうか？

サーバー系 OS（= 汎用 OS）

- Linux に代表される汎用 OS
  - マルチタスク（並列性）
  - マルチコア（スケーラビリティ）
  - マルチユーザー（セキュリティ）
- Web サーバーなど クラウドインフラの基盤 ⇒ コネクテッドサービス基盤
- 各種ネットワークの標準実装
- 車載では IVI, Gateway 領域 で採用中

リアルタイム制御に対応できる RTOS

- Zephyr
  - セーフティ（ISO/IEC 62443)
  - セキュリティ（IEC61508, ISO26262 対応計画中）
- FreeRTOS
- RTEMS (Real-Time Executive for Multiprocessor Systems)
- NuttX (Apache NuttX RTOS)

IT 領域で OSS 採用が先行しましたが、今後は制御系にも広がると期待されています

OSSの “保証” に関わる誤解を解く

“商用 SW には保証があるが、OSS は無保証である”

［保障］OSS ライセンスには 明確に「無保証」と書かれています

GPL v2 11 項「無保証について」 の例

- 『プログラム』は 代価無しに利用が許可されるので、適切な法が認める限りにおいて、『プログラム』に関する いかなる保証も存在しない。書面で別に述べる場合を除いて、著作権者、またはその他の団体は、『プログラム』を、表明されたか言外にかは問わず、商業的適性 を保証するほのめかしやある 特定の目的への適合性 (に限られない) を含む 一切の保証無しに「あるがまま」で提供 する。『プログラム』の質と性能に関するリスクのすべてはあなたに帰属する。『プログラム』に欠陥があると判明した場合、あなたは必要な保守点検や補修、修正に要するコストのすべてを引き受けることになる。

OSS を利用して不具合にぶつかった場合でも”OSS 開発者”には改修を要求できません

［保障］無保証の OSS を利用した製品を なぜ保障できるのでしょうか？

「OSS の無保証」と「製品の保証」はレイヤーが異なります

- OSS の基本となる設計思想は 最適化より汎用性の重視 です
  - 幅広い実行環境（コア数、メモリーサイズ）をサポートします
  - 長期間にわたりコードを継続進化 させています
  - Linux kernel には 30 年以上前のコードも含まれます
  - kernel 内コードは 互換性を維持しないケースもあります
- OSS を利用した場合でも 製品の動作保証 が求められます
  - メーカーは 製品の動作条件（ユースケース） を規定します
  - 動作条件に基づいた システム検証 によって製品を保証します
  - 「OSS ライセンス文」と「保証書」の両方が添付 されます

製品としての "保証"

セットメーカーの開発作業

- インテグレーション
- ユースケース設定
- **システム検証**
- Q&A サポート
- セキュリティパッチ対応

OSS ソフトウエア

- "無保証"
- AS-IS 提供
- サポート義務なし

無保証の OSS 使って、保証のある製品を作れるかが「メーカの腕の見せ所」です

［保障］有償の商用ソフトウエアでも保証がない ケースは多いのです

2024 年 4 月更新

マイクロソフト ソフトウェア ライセンス条項

WINDOWS IOT ENTERPRISE (すべてのエディション)

無保証

お客様のデバイスのソフトウェア (アプリを含みます) は、「現状有姿のまま」で使用許諾されます。お客様の地域の法令により最大限認められる範囲において、本ソフトウェアの品質および性能に関するすべての危険は、お客様が負担するものとします。本ソフトウェアに瑕疵があることが判明した場合、お客様はすべての修正等にかかる総費用を負担するものとします。デバイス製造業者とマイクロソフトのいずれも、本ソフトウェアについていかなる明示的な保証または条件も負いません。製造業者およびマイクロソフトは、お客様の地域の法令により認められる範囲において、商品性、品質、特定目的に対する適合性、侵害の不存在に関するものを含め、黙示の保証、条件、その他の責任を一切負いません。本ライセンス条項では変更できない地域の法令による追加の消費者の権利または法定保証が存在する場合があります。

お客様の地域の法令により、契約上の制限にかかわらず保証、条件、その他の責任を負う必要がある場合、その有効期間は、最初のユーザーが本ソフトウェアを取得後 90 日間に制限されます。製造業者またはマイクロソフトが当該保証、条件、その他の責任を負う場合、製造業者またはマイクロソフトは、自らの選択において、(I) 無償で本ソフトウェアを修理もしくは交換するか、または (II) 本ソフトウェア (もしくは自らの選択により、本ソフトウェアがインストールされたデバイス) の返品を受け入れて購入金額を払い戻します。以上が、お客様の地域の法令に基づく保証、条件、その他の責任に対するお客様の唯一の権利となります。

https://learn.microsoft.com/ja-jp/windows/iot/iot-enterprise/eula/license_ja-jp_japanese_japan.pdf

OSSの “保守” 関わる誤解を解く

”OSS にコードを公開したら、後はコミュニティが保守してくれる”

［保守］コミュニティが 勝手に保守してくれるわけではありません

開発コミュニティが保守している内容

- stable-release（マイナー更新）
  - 期間限定 で提供されます
  - バグ対策
  - 脆弱性 (CVE) 対策
- メジャーバージョンアップには
  - 新機能追加
  - 新デバイス対応
  - 性能改善（内部構造の刷新）
  - 非推奨 となった認証機構等の 削除

開発者（＝パッチ投稿者）が保守する内容

- バージョンアップ時
  - メンテナーからの要求に応じて 内部構造の変更に対応 したコードを提供
  - 新しいデバイス の追加を要求
  - 投稿済みコードの改善（バグ対策）
- バージョンアップに追従しないコードは当面放置され、何れ削除されます
- 「パッチ投稿時に大騒ぎし、マージされたら以降放置」が一番嫌われます

OSS 開発者や企業には「OSS の進化に追従」することが期待されているのです

［保守］公開される CVE = 共通脆弱性識別子への対応 が必要です

https://www.cve.org/

OSS の CVE が多いのは、OSS が脆弱だからではなく採用数が一番多いからです

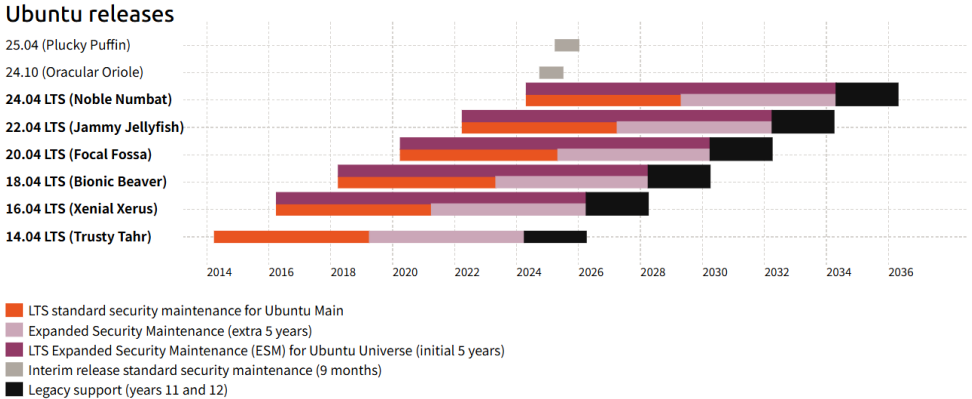
［保守］OSS の 公開された脆弱性情報への対応状況

近年 OSS 開発プロジェクトは サイバーセキュリティ対策に力を入れています

- コミュニティの限定メンバーに 脆弱性情報の公開前に 情報が提供されています
- 社会インフラ等には 情報公開前に対策パッチが提供 されます（ZeroDay 対策）
- CVE 番号が付かない stable-update にも脆弱性対策パッチが含まれているので、CVE 付きパッチのチェリーピックは NG です
- コードをローカルに改変してしまうと、脆弱性パッチが適用できなくなります
- コミュニティによる脆弱性保守期間は比較的短く、出荷済みでも途中でバージョンアップすることが推奨されています（が、出荷済み製品の SW 更新は困難です）
- この解決策として 脆弱性対策パッチを有償で長期提供 する企業もあります

重要な社会インフラ基盤となった OSS には最大級の脆弱性対応が適用されています

［保守］（参考）Ubuntu のセキュリティパッチ提供スキーム



https://ubuntu.com/about/release-cycle

製品開発に OSS を利用する（応用編）

メーカーは 何故製品開発に OSS を利用したい と思うのか

製品の差異化にはならない「必須の共通技術」 が OSS に結集されているからです

- 費用面 のベネフィット
  - 共通技術（ネットワーク、Web、認証、ファイルシステム等）が充実しています
  - 導入初期費用 がかからない ⇒ 研究開発やトライアル（POC）が容易です
  - 量産時の ロイヤリティ費用 がかかりません
- 技術面 のベネフィット
  - 先進性 ⇒ 最先端技術（各種規格、方式、AI モデル）を一番早く利用できます
  - 互換性 ⇒ ネットワーク接続性、ファイルシステム等のデファクトになっています
- 運用面（保守性の良さ）
  - ソースコードへのアクセス が確約されており、脆弱性対策 も提供されます
  - 集合知 なので 生成 AI の得意領域、更に複数の 技術サポート可能な企業 があります

ネットワーク接続やクラウド連携が必要な場合「OSS 活用が第一選択肢」となります

OSS 採用時の「ステータスの見極め」と「社内で確認すべき事項」

不適切な OSS を採用してしまった結果、製品が発売できなかったケースもあります

- 受け入れ可能配布ライセンスか？（他の SW とライセンスが競合しないか？）
- OSS 開発プロジェクトの健全性 の確認
  - 開発プロジェクトに 今でも活動の実体があるか？（ライフサイクルの確認）
  - プロジェクトが 特定企業によって実質的に支配されていないか?
- 社内で 類似 OSS の導入事例 の確認
  - バージョン が一致しているか、適切なバージョンを選んでいるか？
  - 入手元やサポート体制 の確認
- 製品に使った場合に必要となる 長期メンテナンス の確認
  - コミュニティによる保守体制の確認（LTS スキーム があるか）
  - 有償の長期延長サポート の選択肢があるか？
- 製品開発時に必要となる 技術サポートを提供してくれる会社 があるか？

製品開発に OSS を使う時に 絶対にやってはいけないこと

製品開発では ライセンス以外の考慮も必要 です

- ライセンスは任意のコード改変を認めるが...
- Upstream のコードから乖離（fork）しない
- OSS を独自に拡張や改造して製品を作ると
  - 将来の OSS の 更新に追従できない
  - コミュニティの 脆弱性対策が適用できない
  - ネットや 生成 AI の集合知が使えなくなる
  - 結果的に 保守コストが膨大になる
- 野良 OSS（= 独自移植）の利用は論外 です
- OSS の独自拡張は「技術負債」を生みだす

独自改造を考える前に、コミュニティと一緒に問題解決策できないか模索するべきです

OSS を使うと 頻繁なバージョンアップへの追従 が求められます

多くの OSS は「機能の完成ではなく、時間の経過でバージョンを更新」しているから

- 近年は機能更新以外に 脆弱性（サイバーセキュリティ）対策 も増えています
  - マイナーアップデート：バグ修正、緊急的な脆弱性対策の提供
  - メジャーアップデート：新機能の追加、過去の脆弱性対策の累積
- OSS 毎に異なるが、商用 SW より高頻度で更新 されます
  - Linux kernel: メジャー＝原則 9 週間、マイナー＝数日単位
  - Android: メジャー 1 年周期、QPR = 3 か月、マイナー＝毎月
  - Zephyr: 4 か月周期（メジャー、マイナーの区別なし）
- マイナー更新による脆弱性パッチ提供は期間限定 です（次回メジャー更新まで等）
- 製品開発には 長期メンテナンスバージョン（LTS 版）を利用 するべきです

脆弱性対策が義務化される中「脆弱性対策の提供」がメーカの重荷になっています

メーカーの OSS 利活用ステージ ①（利用者 = 一方通行）

OSS 製品開発の「入門企業」には、背後にいるコミュニティが見えていません

- OSS の入手方法
  - チップベンダーやボードベンダー から OSS を サンプルコード として入手しますが 既にメンテナンスが終了した 古いバージョン しか提供されないケースもあります
  - コミュニティから 自分でコードをダウンロードできます（Raspberry Pi など）
  - ディストリビューションを利用できます（Ubuntu, Debian, yocto など）

- 技術課題の解決方法
  - チップベンダーやボードベンダー に聞いても サポートされないケースもあります
  - ネット上には さまざまなバージョンの情報が混在しており 参照時は注意が必要です
  - 時間的な余裕があれば、Web 上のトレーニング教材で勉強 できます
  - OSS 技術サポートを提供している会社に 有償サポート契約 もできます

「OSS 開発コミュニティと繋がらない開発」は結果的に遠回りになっています

メーカーの OSS 利活用ステージ ②（活用者 = コミュニティと連携）

OSS 製品開発の「熟練企業」は、開発コミュニティを味方につけています

- OSS 開発プロジェクトを「認知」しています
  - メーリングリスト（ML） に参加し、自分の 課題が議論されていないか確認します
  - git を参照し、最新バージョンで何が更新されたか 確認します
  - Web ページに公開されている 技術情報、開発者会議の資料やビデオを参照します
- OSS 開発プロジェクトとの「連携」を試みます
  - ML の過去会話などをチェックした上で 問題を再現可能な最少のテストコードやテスト環境の情報を付与して ML に質問 を出します. その際にコミュニティの 最新コードを使ってテストした結果も共有 します
  - 回答義務は無いのですが、有意義な内容には応答してくれる可能性が大きいです
  - 開発者会議等に参加 します（技術の習得に加え 人的ネットワーク も構築できます）

コミュニティ内の議論内容を確認し、自分のテスト結果などをフィードバックします

メーカーの OSS 利活用ステージ ③（事業戦略としてコミュニティ活動）

コミュニティパワーの活用を狙い「全社取り組みとして OSS 開発プロジェクトに参画」

- 開発面
  - 自社エンジニアを 専従者としてプロジェクトにアサインし 共創開発に従事させます
  - 共創開発への参加を通じて コミュニティ内で実績を積み発言力を高めます
  - 開発ロードマップ策定にも影響力を行使 し、自社の 技術戦略と整合 させます
  - 自社の課題解決に 他社の開発者のエンジニアの知見も有効活用 します
- 戦略面
  - 競争優位（最先端技術の取り込みにより 業界への影響力を強化 します）
  - プラットフォーム支配（デファクト技術の獲得によって 業界の囲い込み を図ります）
  - 開発者エコシステム確保（世界中から優秀な OSS 開発者を獲得 します）
  - 自社提案技術を デファクトスタンダード化 し、業界標準を実質的に支配 します

GAFAM 等は OSS 開発を「自社技術開発ロードマップのコア」と捉えています

OSPO（Open Source Program Office）開設は 事業戦略化した証 です

Google オープンソースについて

オープンソースは Google の中核であり、オープンソースを基盤としてあらゆる活動の根幹を担っています。

OSPO の歴史

Google オープンソース プログラム オフィスの歴史は 2004 年に遡り、業界初の OSPO の 1 つとなります。

オープンソースによりオープンに

OSPO は当初、Google がオープンソース テクノロジーを基に構築し、Google が開発したテクノロジーをオープン ライセンスの下で共有できるようにすることに重点を置きました。2005 年に Google Summer of Code が始まったことで、OSPO はオープンソースのメンターシップをサポートし、オープンソースへの参加の障壁を減らすべく拡大しました。現在、OSPO はオープンソース エコシステム全体の改善を目的とした複数のプログラムを実施しています。その中には、Season of Docs プログラム、オープンソース セキュリティの改善を目的とした複数の取り組み、丁寧な言語ガイダンス、行動規範の施行に関するトレーニングなどが含まれています。

世界中の人と共有

OSPO の重要な使命は、Google 主導のプロジェクトの成功と成長を支援することです。2008 年には、Google の注目すべきオープンソース プロジェクトが 2008 年にリリースされました。たとえば、世界各国にスマートフォン向けのオペレーティング システムを提供する Android、Chrome ブラウザが動作する Chromium、効率的なプログラミング言語の Go などが、短作で開発されました。Google は Linux Foundation と提携して 2018 年に Cloud Native Computing Foundation を設立し、現在では Kubernetes などの主要プロジェクトをホストしています。2022 年、Google は IstioMesh を CNCF に寄付し、Kubeflow のインキュベーション プロジェクトにも申請しました。

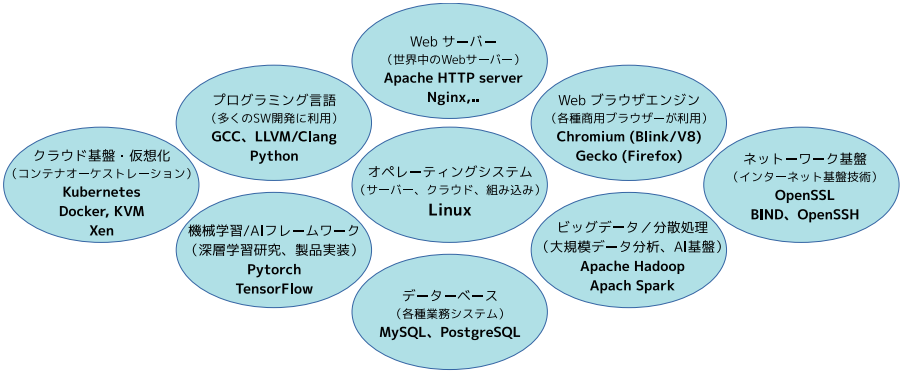
オープンソースの変更に伴う変化

2004 年に Google の OSPO が始まって以来、オープンソースにおいて多くの変化がありましたが、オープンソース エコシステム全体にわたってオープンソース プロジェクト、コミュニティ、管理者をサポートするという Google の取り組みは変わらず変わりません。

https://opensource.google/about?hl=ja

OSS を巡る世界の話題を俯瞰する（上級編）

（再掲）私達の日常はもはや「OSS 無しでは成り立たなくなっています」



重要な社会インフラを支える SW の中には、当然多くの OSS が含まれています

[2021-05] コロニアル・パイプライン社への ランサムウェア攻撃

「SW 脆弱性が国家レベルのリスク要因になりうる」ことを知らしめた大事件でした

- アメリカ東海岸の燃料供給の 45% を担う重要な石油パイプライン 運営会社
- 2021-5-7 にランサムウエア攻撃を受け、一週間にわたり操業が停止しました
- ガソリン、ディーゼル、ジェット燃料などの貯蔵庫が大きな影響を受けました
- 事態を重視した政府は 2021-5-14 にサイバーセキュリティ強化のための 大統領令 に署名、サイバーセキュリティ分野での官民連携 を強化するよう指示しました
- このサイバー攻撃が OSS 対象かは言及されていないが、OSS 開発プロジェクトを多数ホストする Linux Foundation が米議会に呼ばれて 対策を議論しています
- OSS のセキュリティを強化する取り組みの OpenSSF の活動が強化されています

OSS を含めた SW の品質改善、脆弱性対策が「国家の重要戦略」に位置づけられました

[2021-05] Biden 大統領（当時）が 大統領令（14028）を発行しました

Federal Register
Vol. 86, No. 93
Monday, May 17, 2021

Presidential Documents

Title 3—
The President

Executive Order 14028 of May 12, 2021

Improving the Nation's Cybersecurity

By the authority vested in me as President by the Constitution and the laws of the United States of America, it is hereby ordered as follows:

Section 1. *Policy*. The United States faces persistent and increasingly sophisticated malicious cyber campaigns that threaten the public sector, the private sector, and ultimately the American people's security and privacy. The Federal Government must improve its efforts to identify, deter, protect against, detect, and respond to these actions and actors. The Federal Government must also carefully examine what occurred during any major cyber incident and apply lessons learned. But cybersecurity requires more than government action. Protecting our Nation from malicious cyber actors requires the Federal Government to partner with the private sector. The private sector must adapt to the continuously changing threat environment, ensure its products are built and operate securely, and partner with the Federal Government to foster a more secure cyberspace. In the end, the trust we place in our digital infrastructure should be proportional to how trustworthy and transparent that infrastructure is, and to the consequences we will incur if that trust is misplaced.

https://www.govinfo.gov/content/pkg/FR-2021-05-17/pdf/2021-10460.pdf

[2025-06] Trump もほぼ同じ内容で 継続しています（13694、14144）

https://www.whitehouse.gov/presidential-actions/2025/06/sustaining-select-efforts-to-strengthen-the-nations-cybersecu

[2023/2024] 商用サービスで OSS が使えなくなった「リライセンス」

Terraform（HashiCorp）2023-08

- インフラ構成管理ツール
- MPL 2.0 → Business Source License (BSL 1.1) へ変更
- 一般的な使用には無料だが「競合する SaaS プロダクトでの利用」を制限
- 2023-09 OpenTofu として fork
- コミュニティ主体の開発に移行
- インフラ自動化のオープンな代替に

Redis（Redis Inc.）2024-03

- インメモリ DB（Key-Value Store）
- BSD 3-Clause → Redis Source Available License (RSAL) + Server Side Public License (SSPL) に変更
- 非商用利用は無料だが、Redis を使った商用クラウドサービスは提供不可
- 2024-03 Valkey という fork が誕生
- クラウド SaaS が Valkey への移行中

リライセンス対抗策として「元プロジェクトの fork」が確立されつつあります

[2024-03]「xz プロジェクト乗っ取り → バックドア仕込み」事件

xz-utils は xz コマンドなどで広く使われている OSS の 高性能圧縮ライブラリ です

- Jia Tanという人物が 2021 年ごろから xz プロジェクトにメール投稿を開始 し、徐々にメンテナーに信頼され、2022 年以降は 実質的な共同メンテナー に昇格
- 日々の メンテナンス作業の負担で疲弊 していたメンテナーの Collin 氏に対し、Jia Tan は「手伝う」と言って 実質的にプロジェクトの権限を掌握 してしまいます
- Jia Tan は、コードに特定のキーや条件を満たすと SSH 認証がバイパス可能となる高度に難読化された OpenSSH バイパス用のバックドア を仕込みました
- Andres Freund 氏が ssh の異常な CPU 使用率に気付き原因を追及 した結果、xz 5.6.0 の liblzma に疑わしいコードとオブジェクトの改変を発見しました
- この一連の経緯が 2024 年 3 月 29 日、OSS 界に衝撃をもって報告 されました

「悪意を持った開発者」から如何に OSS を守るかも考えなければならなくなりました

[2024-05] EU CRA（Cyber Resilience Act）最終承認、2027 年施行

OSS を利用した製品を開発したメーカーも CRA の規制対象になります

- OSS 開発者（個人、NPO）は、以下の条件を満たせば CRA の規制対象外です
  - ソフトウェアが非営利目的で提供されている
  - ソースコードが公開されており、無料でアクセス可能
  - 提供者が商用的な利益を得ていない
- OSS を利用して製品を開発した製造者（メーカー・輸入業者・販売者）は CRA の規制対象となります
  - 製品開発時のセキュリティ対策
  - 製品販売後のサポート義務（含、最低 5 年間のセキュリティ対策パッチの提供義務）
  - 適合性評価・文書提出義務

違反すると全世界年間売上高の最大 2.5% か 1,500 万ユーロの罰金が課せられます

[2024-10] OSS 開発コミュニティにも 国際情勢が波及 し始めました

これまでは、国籍等に制約されない「個人としての活動の自由」があったのですが...

- Greg Kroah-Hartman が「various compliance requirements（さまざまなコンプライアンス要件のため）」ロシア人とみられるメンテナー数名を登録解除するパッチを提出しました. これで開発済のコードが削除されたわけではありません
- Linus Torvalds は LKML（Linux Kernel Mailing List）にて「これは取り消されることはない」と発言し、Greg の行為を擁護しています
- James Bottomley が後に補足説明として「米国財務省の OFAC（Office of Foreign Assets Control）の SDN（特別指定国民）リストや、その関係企業に所属していた場合、維持管理者リストに名を連ねることが難しい」と背景を解説しました. リストから外れれば個人として復権することができる可能性があるとも言いました

ミッションクリティカル領域に入った結果「政治的な規制」を受けるようになりました

（再掲）こんどは "違和感" が感じられますか？

デジタル・ガバメント推進標準ガイドライン 実践ガイドブック
2025 年（令和 7 年）5 月 27 日 デジタル庁

オープンソースソフトウェアの特徴を理解して採用する

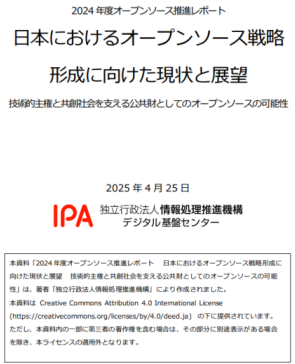
オープンソースソフトウェア（OSS）には、先進的な機能が利用できるメリットがある一方で、不具合があってもサポートを受けられないなどのデメリットもあります。メリットとデメリットの両方を正しく理解した上で、プロジェクトの特性に合わせて、OSS の採用を検討しましょう

メリット		デメリット	
拡張性	・ 公開されているソースコードをもとに、不具合の修正や機能拡張などを行うことができる。	コンプライアンス	・ OSS を利用して独自に開発したアプリケーションについてもソースコードを開示する義務が生じる可能性がある。 ・ OSS 開発者へ損害賠償請求等ができない。 ・ ライセンス違反を理由に第三者から訴訟を起こされる可能性がある。
コスト（※）	・ ライセンス料がかからず、導入コストを抑えられる。 ・ ベースとなる機能や部品として利用することで、開発工数を削減できる。		
先進性	・ 先進的な機能が利用できることも多い。	サポート	・ 緊急時のサポートを受けられない。
セキュリティ	・ 市販のソフトウェア等では、ソースコードを確認することができないが、OSS では、ソースコードが公開されており、脆弱性等を直接確認することができる。	セキュリティ	・ ソースコードが公開されているため、脆弱性を突いた攻撃を受ける可能性がある。
品質	・ 多くのユーザが利用しており、活動が活発な OSS の場合は安定した品質を期待できる。	不具合修正	・ 活動が停滞している OSS の場合、不具合対応されない場合がある。

OSS ではあるものの、製品自体が有償化されていたり、OSS の入手は無償であってもサポートなどが有償化されていたりする場合があるため、OSS の採用を検討する際にコストを確認することが重要です。また、以下の理由で、管理コストが割高になる可能性があることに注意が必要です。
※ OSS はサポート期間が一般的に短いものが多いため、バージョンアップなどの対応が増える場合があります。

https://www.digital.go.jp/assets/contents/node/basic_page/field_ref_resources/e2a06143-ed29-4f1d-9c31-0f06fca67afc/ae9

IPA の問題提起： 私の「違和感」がクリアに解説されていました



2024 年度オープンソース推進レポート

日本におけるオープンソース戦略

形成に向けた現状と展望

技術的主権と共創社会を支える公共財としてのオープンソースの可能性

2025 年 4 月 25 日

IPA 独立行政法人 情報処理推進機構
デジタル基盤センター

本資料「2024 年度オープンソース推進レポート　日本におけるオープンソース戦略形成に向けた現状と展望　技術的主権と共創社会を支える公共財としてのオープンソースの可能性」は、著者「独立行政法人情報処理推進機構」により作成されました。
本資料は Creative Commons Attribution 4.0 International License (https://creativecommons.org/licenses/by/4.0/deed.ja) の下に提供されています。ただし、本資料内の一部に第三者の著作権を含む場合は、その部分に別途表示がある場合を除き、本ライセンスの適用外となります。

https://www.ipa.go.jp/digital/kaihatsu/oss/about/report2024/index.html

CC BY 4.0

このレポートの冒頭部分

- 世界では、オープンソースが公共インフラの一部として制度化され、技術的自立と民主的な技術開発の基盤として活用されている．一方、日本ではいまだ「国家戦略」としての明確な枠組みは存在せず、民間・行政・コミュニティの間で役割と責任の所在が曖昧なまま、個別的な取り組みにとどまっている．
- 本レポートでは、オープンソースを「公共財」として捉え直し、オープンソースに関する国内外の動向をもとに技術的主権の確保と共創社会の実現に向けた国家的戦略の構築を提言する．OSS は無料の道具ではなく、私たち全員が担い手となるべき、社会的な資産である。今こそ、企業・行政・市民が連携し、日本国内を起点としたオープンソースエコシステムを育てる第一歩を踏み出す時だ．

https://www.ipa.go.jp/digital/kaihatsu/oss/about/report2024/index.html

今日のまとめ（Takeaway）

- OSS は、品質・コスト・保守性の面で 十分に製品開発に適用可能 な技術です.
- ただし、ライセンス遵守 や Upstream 追従など「正しい付き合い方」 を理解する必要があります. 特に OSS コードの独自改造（fork）は SW 技術負債の原因 となるので、社内でコードを抱え込まずに Upstream に戻していくUpstream First の考え方が非常に重要です.
- 製品開発に OSS を使うには、技術選定だけでなくメンテナンス・脆弱性対応まで含めた企業レベルでの戦略的な関与 が求められます.

OSS を「無料の部品」と誤解せず、未来の共創基盤と捉える視点が重要です