THE LINUX FOUNDATION | Research

Linux 35

Linux 35 年の歩み

グローバルインフラとイノベーションを支えるオープンソースの原動力

Anna Hermansen, The Linux Foundation
Hilary Carter, The Linux Foundation
Daniel German, University of Victoria

序文：Dirk Hohndel,
Ericsson Software Technology

日本語版翻訳：工内隆・伊達政広

発行：2026 年 10 月



Linux 35 年の歩み

今日、Linuxは40億台のAndroidデバイス、年間3,000万台のChromebook、2億台以上のサーバー、スーパーコンピューターのトップ500のすべて、そしてAppleの5Gモデムを支えています。



Linuxは自動車、世界中の証券取引所、家庭用電化製品、エレベーターやコンベアベルトなどの産業機器、さらには人工衛星や宇宙インフラを動かしています。



Linuxカーネルには1時間あたり約10件の変更が加えられ、8〜9週間ごとに新しいリリースが公開されています。





カーネルは引き続き新たな開発者を惹きつけており、7.0〜7.2の各リリースでは、初めてコントリビューションを行った開発者の数で記録を更新しています。

2026年の初めの2四半期には、各四半期で4,200人を超えるコントリビューターが参加し、継続的な改善を続ける、活発な開発者層が確保されています。

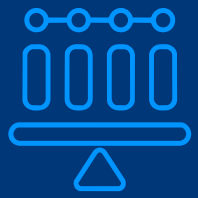




2026年半ばの時点で、131人の個人のコントリビューターがコントリビューション全体の51%。また、7つの組織が51%を占めていました。これはコントリビューターの多様性の高さとレジリエントな開発プロセスを示しています。

現在、全Linuxのコントリビューションの80%以上は企業のエンジニアによるものであり、Google、Ericsson、Fujitsu、Intel、Meta、Huawei、Samsung、SUSEなどの企業が協調して取り組んでいます。





2026年のコントリビューターの50%以上が複数四半期にわたって継続的にコントリビューションしており、コミュニティは安定した、参加意欲が高い状態にあることを示しています。

カーネルのソースコード開発は、「バージョン管理導入前」、「BitKeeperの時代」、「Gitの時代」という3つの段階を経て発展してきました。こうした移行のたびに、プロジェクトはより複雑かつ成熟したものになって行きました。



Linuxは、レビュー主導型のコラボレーションを先駆的に導入し、開発における透明性の高い分散型ワークフローであるGitを生み出すことで、現代のソフトウェアエンジニアリングに革命をもたらしました。



Linuxは事実上すべての主要な視覚効果（VFX）レンダーファームを支えており、ValveのProton互換レイアを通して異なるアーキテクチャ間でのPCゲームプレイを可能にしていています。



世界の証券取引所や銀行は、低遅延取引、リスクモデリング、そしてオープンソース サプライチェーンの強靭性確保といった目的で、Linuxを活用しています。



Copyright © 2026 **The Linux Foundation** | 2026 年 10 月 . 本レポートは、**Creative Commons Attribution-NoDerivatives 4.0 International Public License** の下でライセンスされています。
インフォグラフィックを使用する際は、本調査レポートを引用してください。引用に関するガイダンスは、レポートの裏表紙をご覧ください。

Linux 35 年の歩み

Linuxは、チップメーカー各社に対して、Wi-Fi関連の規格をオープンにするよう促しました。その結果、Linuxは、世界中のセルラーモデム、家庭用ルーター、そして企業向けアクセスポイントを支える基盤へと成長しました。



ヘルスケア分野では、組み込みLinuxが国際的な安全認証規格を遵守しながら、医療機器、診断ツール、ゲノム研究を強力に支えています。



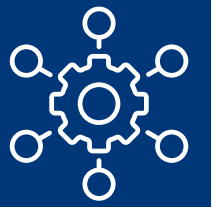
自律型宇宙ミッション、衛星コンステレーション、商用飛行ドローンなどでは、リアルタイムの航法アルゴリズムを実行し、複雑なテレメトリ処理を行うためにLinuxが利用されています。





Automotive Grade Linux（AGL）は、安全重視のソフトウェアに対する長期サポートを備えた協調的フレームワークのもとで統合された車両ソフトウェアスタックを提供しています。

Linuxの安定性、カスタマイズ性、拡張性は商用プロプライエタリー市場を大きく変革し、現在では主要なWebサイト、メインフレーム、サーバー、モバイルデバイスの大部分がLinuxによって動作しています。





クラウドネイティブのイノベーションは、Linuxカーネルに直接組み込まれた中核的なプリミティブによって実現され、それらは現代のソフトウェアワークロードに欠かせないランタイムとなっています。

リアルタイムカーネル機能とハードウェアポータビリティのおかげで、Linuxはローカルエッジデバイス、地域データセンター、5G基地局の領域を席巻しています。





Linuxは、計算負荷の高い並列化されたAIワークロードを支えるべく進化を遂げてきました。それと同時に、AI関連のツール群は、カーネルの保守、パッチのレビュー、バグの検出といったプロセスを大きく変革しました。

Linuxカーネルの非公開セキュリティ メーリングリストにAIが生成したパッチが大量に投稿されたことを受け、脆弱性報告の基準を明確化し、AIボットによる投稿を制限するよう関連文書が更新されました。



Linuxコミュニティは、主要な脆弱性クラスを排除するためにKSPPを立ち上げて予防的な構造型防御を組み込むとともに、メモリ安全性を確保するドライバー開発に向けてRustを採用しました。



LF Mentorshipプログラムは、奨励金、ウェビナー、メンターシップを通じて参入障壁に対処し、オープンソースの多様性、技術的スキル、およびソフトウェア工学における「人」の側面を育成しています。



利用しやすいデジタル決済、農業ツール、クリーンエネルギー、グローバルな接続性を可能にすることで、Linuxカーネルは世界中の何百万人もの人々の重要な日常生活を静かに支えています。



Copyright © 2026 **The Linux Foundation** | 2026 年 10 月 . 本レポートは、**Creative Commons Attribution-NoDerivatives 4.0 International Public License** の下でライセンスされています。
インフォグラフィックを使用する際は、本調査レポートを引用してください。引用に関するガイダンスは、レポートの裏表紙をご覧ください。

目次

序文 **5**
失った人々に思いを寄せて 7
はじめに **8**
Linux のアーキテクチャ **9**
Linux カーネル開発の 3 つの時代 10
ソフトウェアエンジニアリング実践における Linux のインパクト 12
産業界における Linux のインパクト **13**
金融サービス 14
通信およびネットワーク 15
自動車 16
映画およびゲーム 17
ヘルスケア 18
製造業と市民社会のインフラ 20
宇宙、ドローン、自律型システム 21
Linux の技術的なインパクト **24**
個人および企業向けコンピューティング 24
クラウド コンピューティングとコンテナ 27
エッジ 29
人工知能 30
持続可能性とカーネルの未来 **33**
セキュリティ 33
次世代のためのメンターシップ 34
カーネルが人々に与えたインパクト **37**
おわりに **38**
謝辞 **39**
著者紹介 **40**
参照資料 **41**

序文

プロジェクトの歴史について語る際、人々は現在を起点として過去を振り返ります。その際、かつて起きた出来事は、必然的に「今日私たちが知っていること」という文脈に照らして解釈されることになります。しかし、Linux が始まった当初、そのプロジェクトがどのような軌跡をたどるかは、誰にも分からず、将来これほどの発展を遂げるとは夢にも思われていませんでした。当初、Linux は Linus Torvalds による単なる個人的なお遊びの実験に過ぎず、386 アセンブラを学び、そのプロセッサのメモリ管理機能を理解するための試みでした。あくまで小さな趣味のプロジェクトであり、ごく少数の人の目に触れる程度で、それも関心を持つ人はさらに少ないだろうと思われていたものに過ぎなかったのです。

Linus は大胆な発言を即座に口にすることで知られていますが、それ以上に、彼の言葉が誤って引用されたり、自身の目標に関する発言が誤解されたりすることも多くありました。（有名な例として「世界征服、それも素早く」というものがあります。）しかし実際には、プロジェクト開始から 3 年後の 1994 年に Linux カーネルのバージョン 1.0 をリリースした時点でも、彼は Linux を「予想以上に多くの人々を惹きつけることになった小さな趣味のプロジェクト」と考えていました。

そのマインドセットこそが初期の成功に大きく寄与しました。そして、その成功を維持可能にしたのは、Linus が時代とともにこのマインドセットを変化させ、機能しているものから学び、最も重要なこととして「機能しないものがあるときには進んで変更を加える」能力でした。時には非常に大きな変更を行うこともありました。たとえば、カーネルのリリースモデルを完全に新しい手法へ変更すると決断したときがそうです。当時は正直なところ、周囲から不可能だと広く受け止められていました。9 〜 10 週間ごとに新しいリリースを行う？すべての作業が従うべき、容赦なく予測可能なペースを確立する？リリース計画も、壮大なビジョンも、数ヶ月かけて綿密に練り上げられたプロセスも存在しない――。多くの人はこのアプローチがすぐに失敗すると予想しました。しかし 20 年以上経った今でもそのプロセスが使用されており、それこそが Linux の成功の大きな理由となっています。2005 年当時のその決断は、単に失敗した旧プロセスへの対処であり、根本的な変更を行い、失敗から学び、前進しようという意欲の表れだったのです。

そして同じパターンが何度も繰り返されました。上手くいっていることを続け、上手くいかないことを疑問視し、問題に対処するための変更を進んで行うことです。未来を予測できる必要はありません。昨日よりも明日を良くする方法を見つけ出すことができれば良いのです。

あるいは、Linus 自身の言葉を借りれば、こうです。「私は先見の明があるような人間ではありません。5 カ年計画なんてものも持っていません。私はエンジニアです。……足元を見て、目の前にある穴に落ちないよう、それを塞ぎたいと思う。私はそういう人間です。」

Linux の 35 年という歴史を語る際、私たちは、その驚くべき「粘り強さ」（「頑固さ」と言うより、ずっと聞こえがいいでしょう。）に注目すべきです。この「粘り強さ」があったからこそ、数々の小さな決断が積み重なり、単なる趣味のプロジェクトに過ぎなかったものが、史上最も重要かつ影響力のあるソフトウェアプロジェクトの一つへと成長したのです。

Dirk Hohndel
Head of Ericsson Software Technology



Linus Torvalds と Dirk Hohndel が Open Source Summit North America, Minneapolis にて講演、2026 年 5 月

失った人々に思いを寄せて

35 年間にわたり、Linux カーネルは共通の試みに知識、時間、アイディアを持ち寄った人々の共同の努力によって成長してきました。メーリングリストを通じて交わされたコントリビューションはコラボレーションとなり、コラボレーションは友情へと変わり、代々の開発者たちが、自らを教え、励まし、高め合える仲間を見出してきました。

このコミュニティの形成を助けた人々の中には、すでに世を去った人々もいます。

彼らの業績はカーネルの一部として残り、その上に構築されたシステムを形作り続けています。しかし、彼らが残した遺産はソフトウェアそのものを超えています。カーネルコミュニティの強さは常に人々のつながりに依存してきました。難しい問題を説明し、誰かが道を見つけるのを助け、新鮮な視点を提供し、作業にユーモアと温かさをもたらしてくれる人々、そうした人々がコミュニティを支えてきたのです。

そうした関係は永続的な影響を与えます。開発者たちは技術的な知識だけでなく、どのように協力し、どのように難しい問題にアプローチし、どのように次の人が成功するのを助けるかという教訓を未来へ継承します。このようにして、先人たちの影響力は彼らが教え、支え、インスピレーションを与えた人々を通じて生き続けています。

彼らがいないという事実は、コミュニティとは、それぞれが独自の「声」と「存在感」を持つ個人から成り立っているのだということを私たちに気づかせてくれます。彼らが加わっていたはずの会話、彼らが答えてくれたはずの問い、そして彼らがもたらしてくれたはずの視点——それらが失われたことを、私たちは寂しく感じています。私たちは、コードの背後にいた彼らを——その好奇心、寛大さ、ユーモア、意志の強さ、そして友情を——恋しく思うのです。

Linux の 35 周年を迎えるにあたり、私たちは、すでに亡くなられたコミュニティのメンバーに思いを馳せます。彼らがカーネルやオープンソースにもたらしたコントリビューションを称えるとともに、コミュニティに対して彼らが捧げてくれたもの——共有された知識、育まれた友情、築き上げられた伝統、そして、他の人々が活動を続けていく支えとなった数え切れないほどの小さな支援の数々——に深い敬意を表します。

カーネルは、新たな世代に受け継がれながら、これからも進化し続けて行きます。先人たちの功績は、その続いていく物語の中に織り込まれており、彼らを知る人々に与えた影響もまた、そこに息づいています。

私たちは、感謝と親愛の念を込めて彼らを偲びます。彼らが Linux にもたらしたもの、そして互いに分かち合ったものに対して心から感謝するとともに、今もなお、彼らがいないことの寂しさを感じています。

はじめに

35 年前、Helsinki University の Linus Torvalds は、世界を確実に変えることになる技術を世に送り出しました。それが Linux カーネルです。当時、このオペレーティングシステム カーネルの趣味のプロジェクトが、世界のデジタルインフラを支える存在にまで成長すると予想できた者は誰もおらず、生みの親である Torvalds 自身でさえ想像すらしていませんでした。今日、このカーネルはあらゆる産業の原動力となり、新たな企業の誕生や製品の創出を可能にし、さらには史上最大級のソフトウェア協調開発プロジェクトを支えるに至っています。

「なぜ Linux カーネルは、ソフトウェアとしてこれほど重要なのでしょうか?」と、Michael Dolan (Linux Foundation の Legal and Strategic Programs 部門 senior vice president) は問いかけます。「それは、世界中のソフトウェアの大部分が対応している、実績あるオペレーティングシステム基盤として、誰でもいつでも即座に何かを構築できるからです」[1]。Linux カーネルのように強力な技術がオープンなライセンスで提供されたことは、また現在も提供されていることは、その普及を大きく加速させる原動力となってきました。このオープンであることは、同じく Torvalds が開発した、バージョン管理やソフト開発において大規模な協調作業を可能にするツール Git と組み合わさることで、350 社を超える企業から数千人もの開発者がプロジェクトに参加することを可能にしました。また、低コストで手頃なソフトウェアソリューションを開発する手段を提供しているという点で、経済的にも重要な影響をもたらしています。例えば、Android デバイスは手頃な価格で購入できるため、インドにおける Android の市場シェアは 95% を超えています[2]。こうしたアクセスのし易さにより、より多くのインドの人々にオンラインでつながる環境をもたらしているのです。

1991 年の最初のリリース以来、Linux カーネルは広大な開発者コミュニティのコントリビューションを経て成長を続けてきました。現在では、一時間あたり約 10 件の変更が加えられており、予測可能な 8 ～ 9 週間のサイクルで新しいリリースが行われています。Linux が驚異的なイノベーションの道を歩み続けてこられたのは、常に開発者たちの存在があったからです。35 年という歳月が経過した今、他の類似プロジェクトの状況を鑑みて、カーネル開発者コミュニティは縮小しているのではないかと思われるかもしれません。しかし、現実はそうではありません。Jonathan Corbet が指摘しているように、2026 年のカーネル 7.1 リリースには、初めてコントリビューションした 530 名による成果が含まれていました。これは当時記録的な数字でしたが、その記録は長くは続きませんでした。カーネル 7.2 のリリース候補版（RC）の段階で、すでに 613 名もの新規コントリビューターによるパッチが取り込まれているのです[3,4]。

今日、Linux は 40 億台の Android デバイス、過去 10 年間にわたり年間 3,000 万台出荷される Chromebook、Apple の 5G モデム、2 億台以上のサーバー、TOP500 のスーパーコンピューターのすべて、自動車、世界中の証券取引所、家庭用電化製品（テレビ、サーモスタット、洗濯機など）、産業機器（エレベーター、コンベアベルト、風力タービンなど）、さらには人工衛星や宇宙インフラを動かしています。それでも、このリストは決して網羅的なものではありません。Linux カーネル メンテナーの Greg Kroah-Hartman は、「これらの数値の多くは『端数程度の誤差』にすぎません。」と言います。要するに、彼は、「すべてが Linux 上で動いているのです[5]。」と言います。カーネルからは、Docker、eBPF、Device Tree など、ソフトウェアスタックの重要なコンポーネントとなっている他のプロジェクトが派生しています。これらの副産物はカーネル自身の変化の速度に好影響を与えるだけでなく、今後の OS の将来の方向性にも影響を与えています。

本稿では、Linux カーネル プロジェクトの 35 周年を祝うとともに、同プロジェクトにおける重要な開発の節目やアーキテクチャを掘り下げます。その際、産業界やさまざまな技術分野においてプロジェクトが現時点でもたらしている影響と、Linux カーネルを形作り、今後も長期にわたって形作り続けるコントリビューターコミュニティのダイナミクスの双方に焦点を当てています。



Linux のアーキテクチャ

Linux は、オープンソースのコンピューティング システムにおける空白を埋める必要性から生まれました。1983 年、Unix の商業化が進み、制限が強まる中、Richard Stallman は、完全に自由で Unix と互換性のあるシステムを構築するために GNU プロジェクトを立ち上げ、その自由を保護するための GNU General Public License (GPL) を策定しました[6]。1990 年代初頭の時点で、GNU プロジェクトは必要な構成要素の大部分を揃えていたものの、実運用に耐えうるカーネルは欠けていました。1992 年に GPL-2.0 ライセンスを採用した Linus Torvalds の Linux カーネルは、まさにそのミッシング ピースとなり、完全に自由な Unix 系システムを実現可能なものにしたのでした。

Torvalds は Linux カーネルを作成し、自身のプロジェクトに対する開発者からのフィードバックを期待して、カーネルのソースコードを公開しました。あるインタビューによると、彼はこれをオープンソースの取り組みとは見なしていませんでしたが、コントリビューターたちが次々と機能を追加していく中で、このプロジェクトはオープンソース ムーブメントと結びつけられるようになりました[7]。カーネルに加え、共同利用や共同開発を前提としたライセンスを持つソフトウェアの利用が拡大したことが、「オープンソース」という用語が生まれるきっかけとなりました[8]。カーネルへのコントリビューションは 1991 年以降劇的に増加し、2014 年頃には一回のリリースあたり 1 万 2000 ～ 1 万 5000 件のコミットという水準で安定するようになりました[9]。本レポートの執筆時点で、Linus はバージョン 7.2 をリリースしています。このバージョンには 2,652 人の開発者から 16,418 件のコミット（マージコミットを除く）が寄せられ、カーネル史上最多件数を記録しました。切れ目のない継続的なリリースサイクルを軸としたオープンかつ協調型開発プロセスにより、カーネルは時とともに有機的かつ迅速に進化し、成熟してきました。各リリースには、カーネルの有用性を維持するためのさまざまな機能が盛り込まれています。例えば、複数の CPU を搭載したシステムで Linux を動作可能にしたバージョン 2.0 の対称型マルチプロセッシング（SMP）サポート、ハードウェアサポート高度化とスケーラビリティ向上を実現したバージョン 2.6、ライブカーネルパッチを導入したバージョン 4.0、そしてリアルタイム サポートがメインラインに取り込まれたバージョン 6.12 などが挙げられます[10,11]。

広義の Linux アーキテクチャは、さまざまなコンポーネントで構成されており、その中には以下が含まれます。

- ハードウェア層：CPU、RAM、入出力デバイスなど
- Linux カーネル：プロセス、メモリ、ファイル、デバイスを管理し、ハードウェアとアプリケーションの間の通信経路として機能
- ライブラリ：アプリケーションがカーネルと通信するための機能を提供
- シェル、あるいはコマンドライン
- ユーザー空間：アプリケーションが実行される領域 [12]

当初は 32 ビットの Intel x86 マシンでしか動作しなかった Linux を、64 ビット RISC ベースの DEC Alpha マシンへと移植するという Linus の初期の決断により、カーネルを移植可能なものにするための書き換えが必要になりました。これが、時計からスーパーコンピューターに至るまで、あらゆる機器で Linux が動作するポータビリティの土台となりました[13]。この書き換え以降、Linux のアーキテクチャは、多様なハードウェア環境に合わせてカーネルを柔軟に構成できるものとなりました。また、カーネルのさまざまな用途に合わせて開発された Linux ディストリビューションも数百種類存在します。主要なものとしては、Ubuntu や Raspberry Pi OS の基盤となった Debian、Fedora を生み出した Red Hat のエンタープライズ向けディストリビューション、そして Android などが挙げられます。こうした高いモジュラリティにより、今日、Linux は広く普及するに至り、現在、Linux は 2 億台のサーバー、数十億台の Android システムをはじめ、クラウドプラットフォーム、スーパーコンピューター、組み込みシステム、Chromebook、家電製品、自動車、企業インフラ、さらには宇宙船に至るまで、幅広い分野を支えています。

Linux カーネル開発者シリーズ：Jonathan Corbet



Jonathan Corbet がカーネルプログラミングの世界に足を踏み入れたのは、1980 年代初頭、大学在学中のことでした。当時、システム管理の専門家であった Evi Nemeth が、VAX システム上で動作する初期の BSD リリースのソースコードに彼がアクセスできるよう手配し、大規模ジョブが他のプロセスを圧迫しないよう、メモリ管理の仕組みを修正する課題を与えたのです。大学卒業後、彼は National Center for Atmospheric Research（NCAR）で、初期の気象関連「フリー」ソフトウェアに携わり、DECUS（DEC ユーザ会）の配布する磁気テープを通じてコードの作成・配布を行っていました。その後、Linux カーネルがリリースされると、彼はすぐにこれに関心を持ち、NCAR での業務の一環として、まずは X Window System に改良を加えるところから取り組み始めました。やがて彼は NCAR を退職し、Linux の開発に専念するようになりました。

カーネルメンテナーとしての彼のキャリアはそこから発展し、コミュニティの鼓動の中にいるような役割やプロジェクトを担うようになっていきました。「Git の時代」以前、Jonathan が最初に行ったコントリビューションは、Linus Torvalds に直接メールを送り、公開されたパッチセットを調べて、コントリビューションが受け入れられたかどうかを確認するというものでした。1998 年には LWN.net を共同設立しました。同サイトは、カーネル開発やコミュニティの動向を伝える、独立系の主要なニュース媒体へと成長しています。また、書籍『Linux デバイスドライバ (第 3 版、邦訳オライリー・ジャパン)』を共著し、後にはカーネルの公式ドキュメントのメンテナーも務めました。さらに、Technical Advisory Board の最初の選挙で選出されて以来、約 20 年にわたって同委員会に携わり、異なるサブシステムや企業、開発者たちの間をつなぐ役割を果たしました。彼は自身の委員会での役割について、次のように語っています。「おかげで、コミュニティで何が起きているのかをより深く把握できるようになりました。本来対話すべきなのにそうしていない人たち、似たような問題に取り組んでいる人たち、あるいは互いに助け合える可能性のある人たち同士を結びつけようと努めてきました。」

彼がなぜ今もコミュニティで活動を続けているのかについて、彼は次のように語りました。「私たちは皆、『世界をさまざまな形で変えられるはずだ』という思いを抱いて、この取り組みを始めました。当初は、自分たちでコントロールできないプロプライエタリーのソフトウェアやシステムが支配を広げることに対する戦いでした。そして、世界を変えるという点において、私たちは想像をはるかに超える成功を収めました。私たちが作り上げた世界が、当初思い描いていたものと一致しているかどうかは分かりませんが、それこそが『世界を変える』ということの性質なのです。」

Jonathan のインタビュー全編、**Life as a Linux Kernel Developer and Maintainer with Jonathan Corbet** を、是非ご視聴下さい[14]。

28 年もの長きにわたり、Linux カーネルの進化とコミュニティの歩みをたゆむことなく記録し続けてこられた、Jonathan および LWN.net チームの皆様に心から感謝いたします。本レポートの執筆者を含むオープンソース コミュニティは、LWN.net の寄稿者たちによる徹底的かつ緻密な取り組みから多くを学び、また、大きく依拠してきました。

Linux カーネル開発の 3 つの時代

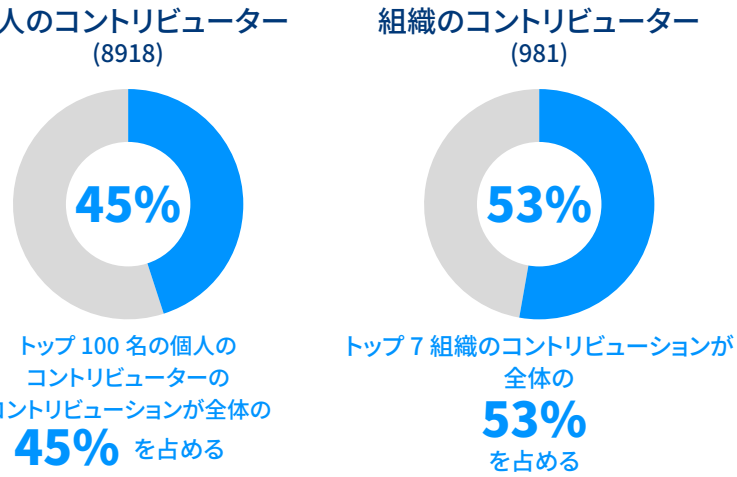
Daniel German が開発したツール cregit は、Linux カーネル開発の状況を詳細に把握できるものであり、カーネルのソースコードの変遷を、「バージョン管理導入前」、「BitKeeper の時代」、「Git の時代」という 3 つの段階に分けて追跡しています[15]。1991 年 9 月から 2002 年 2 月 4 日まで続いた「バージョン管理導入前」の時代には、Linux のリリースそのものだけが、開発規模をうかがい知る手がかりとなっていました。2000 年代初頭までには、Linux は、最初の 10 年間を支えてきたパッチ中心のワークフローの枠を超えて成長していました。カーネルは、多数のサブシステム、メンテナー、アーキテクチャ、ドライバー、そして商業的なステークホルダーを抱える、大規模かつ急速に発展するプロジェクトとなっていました。2002 年 2 月 4 日、分散型バージョン管理システム BitKeeper が導入されました。これは、メンテナーが独立した作業の流れを管理し、履歴を保持し、変更を統合し、大規模な開発を協調させるための、より優れた手段を提供することを目的とするものでした。しかし、開発者にその利用を促すことについて、Torvalds は次のように述べています。「……真の成果が得られるのは、主要な開発者たちも bk（BitKeeper）を使い始めたときだ。……それには時間がかかるだろう」[16]。この開発の第二の時代は、Linux カーネルにおけるコミット履歴の始まりを告げるものとなりました。

この移行は、Linux の開発を加速させました。2003 年末、コミュニティは歴史的なカーネルになる Linux 2.6 をリリースしました。このシリーズは、スケーラビリティ、応答性、thread 機能、I/O、CPU およびハードウェアのサポートが向上し、エンタープライズやサーバーのワークロードにも適していることから、カーネルの歴史における重要な転換点と見なされています。Linux は、現代のコンピューティングインフラの基盤となりつつありました。

この開発プロセスは、システムとともに成熟していきました。BitKeeper は、より分散的で、メンテナーを中心とする協調モデルを強化しました。一方で、Signed-off-by 行の導入や、開発者による Developer Certificate of Origin（DCO）の策定・採用といった取り組みは、アカウンタビリティと変更来歴を確かなものにしました[17]。Linux の開発は、レビュー、責任、そして変更の追跡が可能であるという特徴を維持しつつ、分散的な作業を協調して実施するための体系的なプロセスへと発展しました。BitKeeper が利用されていた期間中、カーネルは 8,740 ファイルから 17,844 ファイルへと拡大しました。

現在のバージョン管理システムである「Git の時代」は、2005 年 4 月 16 日に幕を開けました。Linux コミュニティが BitKeeper を利用するための無償ライセンスを失ったことをきっかけに、Linus Torvalds は新たなバージョン管理ツールの開発に乗り出しました[18]。彼は、高速かつシンプルであること、サポートされること、分散開発に適していること、そして大規模なプロジェクトにも対応できること、といったいくつかの原則を念頭に置いてこのツールを構築しました[19]。また、彼はこのプロジェクトをオープンソース化しました。Git は、カーネルの開発プロセスのみならず、事実上すべてのオープンソースプロジェクトを一変させました。さらに、カーネルにおける、高度に分散された多様かつ健全なコントリビューター層の維持においても、間違いなく極めて重要な役割を果たしてきました（**図 1** を参照）。ソフトウェア エンジニアリングに対する Git の影響については、以下の囲み記事で論じます。

図 1：カーネルコントリビューター統計（2025 年 9 月～ 2026 年 8 月）



2025 年 9 月から 2026 年 8 月までの期間は、カーネルには 8,918 名の個人コントリビューターが積極的に関与しており、そのうち上位 100 名が全コントリビューションの 45% を占めています。また、981 の組織が積極的にコントリビューションしており、そのうちの 7 組織が全コントリビューションの 53% を占めています。コントリビューターの定着率は 55% ～ 75% の範囲でした。

出典：*The Linux Kernel Organization Insights（LFX）。参照元：* https://insights.linuxfoundation.org/project/korg?timeRange=past365days&start=2025-08-28&end=2026-08-28 [20]



2024 年 4 月、シアトルで開催された *Embedded Open Source Summit* の参加者

ソフトウェアエンジニアリング実践におけるLinux のインパクト

Linux のインパクトは、単なるオペレーティングシステムとしての利用にとどまりません。現代のコラボレーション実践の主要な推進役として、Linux はソフトウェアエンジニアリングのあり方を変えました。Linux は、開発がインクリメンタルに進行し、レビュー主導で行われ、バージョン管理されるべきだという考え方を一貫して推し進めてきました。カーネル開発の規模が拡大し、開発者が地理的に分散するようになる中で、Linus Torvalds は、大規模かつ分散したグループがいかにして協調して、ソフトウェアを開発すべきかを再考し、その結果として Git を生み出しました。Git は、カーネル開発のワークフローを軸とした分散型バージョン管理の仕組みを完成させました。すなわち、変更は、提案、レビュー、議論、テスト、および統合といった一連の目に見えるプロセスを通じて行われます。

Git は、カーネル開発における実践手法を、カーネルそのものの枠を超えて広める役割を果たしました。Linux 規模のプロジェクトにおいて、容易かつ高速で分散的な作業を支えるべく設計された Git は、やがて他のオープンソース プロジェクトや、さらには商用ソフトウェア開発チームにも採用されるようになりました。Git を軸としたホスティング プラットフォーム、コードレビュー システム、継続的インテグレーション（CI）ツールなどが発展するにつれ、カーネルコミュニティで磨き上げられた手法は、ソフトウェア開発の日常的なインフラの一部となりました。

> **したがって、Linux のインパクトは、今日のインフラの大部分を支えるシステムというだけにとどまりません。Linux は Git を通じて、現代のソフトウェア開発モデル――すなわち、分散型であり、インクリメンタルで、レビューを重視し、変更の履歴が透明に保たれるという開発モデル――の確立にも貢献したのです[19]。**

産業界における Linux のインパクト

学術的および経済的な影響分析によると、もし Linux のようなオープンソースソフトウェアが突然消滅した場合、運用効率の低下、総所有コスト（TCO）の増大、そして研究コストの重複といった要因により、長期的には各国の GDP が平均して 2% 以上低下する可能性があることが示されています[21]。現在、カーネルへのコントリビューションの 80% 以上は企業所属のエンジニアによるものであり、プロジェクトの開始以来、数千もの組織が少なくとも 1 名以上の活動的なコントリビューターを輩出してきました[20]。 アジアからは Fujitsu、Huawei、Samsung、ヨーロッパからは Ericsson、Igalia、SUSE、そして米国からは Intel、Red Hat、Google、Meta、AMD、NVIDIA、Qualcomm といった世界的な大企業が、数多くの他社組織と共に、このプロジェクトで協力し合っています。2026 年における主要なコントリビューション組織のリストについては、**図 2** を参照してください。

特筆すべきは、Linux がソフトウェア定義産業（software-defined industries）の普遍的な基盤となっていることです[22]。 本節では、Linux が業界特有のイノベーションを形成する上で果たす役割、そして現在、業界共通の課題や企業間競争に入る前のソフトウェア基盤に関する広範な協業について述べています。

> **Linux はソフトウェア定義産業の普遍的な基盤に**

図 2：Linux カーネルへのコントリビューションの多い組織の順位（2025 年 9 月～ 2026 年 8 月）

	組織	総コントリビューション数
1	Linux Kernel	48,192
2	Intel Corporation	32,276
3	Red Hat	17,935
4	Advanced Micro Devices, Inc.	15,491
5	Google	13,786
6	SUSE	12,027
7	Meta Platforms, Inc.	7,422
8	NVIDIA	7,045
9	Linaro	6,912
10	Qualcomm Inc	6,044

各組織の順位は、その組織に所属するコントリビューター（またはその組織の名の下に活動するコントリビューター）によるコントリビューションの数に基づいて決定されています。

出典：*The Linux Kernel Organization Insights（LFX）*。参照元：https://insights.linuxfoundation.org/project/korg?timeRange=past365days&start=2025-08-28&end=2026-08-28[20]

金融サービス

1991 年に Linux カーネルが登場した当時、金融サービス業界は、それが採用・導入される可能性が最も低い分野の一つと見なされていたでしょう。当時の銀行や取引所の業務は、独自の Unix 系 OS やメインフレーム上で処理されていたからです。リスク回避を重視し、厳格な規制の監視下に置かれる金融業界において、コミュニティによって保守される無償のコードを、極めて重要な取引インフラの基盤に据えるという発想は、あり得ないもののように思われました。しかし 1990 年代後半になると、Linux を採用する合理性が急速に浸透し、金融サービス業界は「ミッションクリティカルな業務アプリケーションに Linux を導入した、最初の主要な市場の一つ」となりました[23]。

カーネルの透明性、迅速なパッチ提供サイクル、そしてシステムごとのライセンス料が不要であるという点は、ベンダーロックインを伴うプロプライエタリーな Unix を避けつつ、電子取引やバックオフィス処理の規模を拡大する必要がある業界にとって、Linux を魅力的な選択肢にしました。初期の導入企業は、まず Web サーバーやバッチ処理、テスト環境といった周辺のワークロードに Linux を採用し、その後、データセンターの中核部分へと利用範囲を広げていきました。

この分野における利用事例は急速に拡大しました。取引所や値付け業者は、ネットワークおよびスケジューリングのレベルにおける低遅延性と調整柔軟性を評価して Linux を採用しました。これは、決済や取引に求められるスピードを実現する上で極めて有効でした[24]。投資銀行は、リスク計算やデリバティブの価格算出を行うためのグリッド コンピューティング環境を Linux クラスター上に構築しました。これは、大規模な運用において、商用 Unix よりも低コストかつ柔軟性に優れていたからです。やがて Linux は、ニューヨーク、シカゴ、ロンドン、フランクフルト、上海をはじめとする世界最大級の取引所そのものを支える基盤へと発展しました。各取引所は、従来のインフラと比較した際のパフォーマンスとコスト面での優位性を挙げ、中核となる取引プラットフォームを Linux ベースのシステムへと移行させました[25]。

その影響は、単なるコスト削減にとどまるものではありませんでした。Linux の採用は、「オープンかつ協調体制のもとで保守されるソフトウェアであっても、厳しく規制される分野に求められる信頼性と監査可能であるという要件を満たし得る」という考え方を、当たり前のものとして定着させました。それは、透明性（コードを調査し、修正し、検証できること）が、規制当局の求めるリスク管理と両立するだけでなく、むしろそれを強化し得るものであることを示しました。こうした考え方の転換は、技術そのものと同じくらい重要な意味を持っていました。すなわち、オープンソースは、システム上極めて重要なインフラを任せるに足る信頼性を備え得る、ということを証明したのです。

> **Linux の普及により、オープンかつ協調体制のもとで保守されるソフトウェアが、厳格な規制が求められる分野における信頼性と監査可能であるという要件を満たし得るという考え方を、当たり前のものとして定着させました。**

その先例が、業界における本格的な協調の土台を築きました。各機関・企業がそれぞれ独自に、規制の厳しい環境にオープンソースを適応させるための手法を模索するのではなく、業界全体でそうした取り組みを持ち寄る方向へと進んだのです。この考え方が Symphony Software Foundation の設立につながり、それが後の **FINOS**（Fintech Open Source Foundation）に発展しました。現在、FINOS は Linux Foundation の一部となっています。
今日、このコミュニティには銀行、フィンテック企業、テクノロジーベンダーなど 100 を超える組織が参加し、金融分野における共通のオープンソース ツール、標準規格、ガバナンスのあり方を共同で構築しています。FINOS の Executive Director である Gabriele Columbro は、業界を活性化させる原動力としての Linux の意義について、次のように振り返ります。「35 年前、金融サービスにおける極めて重要なインフラが、誰でも自由に利用できるカーネル上で稼働するなどと予想した人は、ほとんどいなかったでしょう。今日では、まさにそのオープンであるという事実が、取引所やリスク管理システムを支え、業界全体で共通のレジリエンスを築くための協調体制を可能にしています。」



2026 年 6 月、ロンドンで開催された *Open Source in Finance Forum* に登壇している *Gabriele Columbro*

通信およびネットワーク

21 世紀の幕開けとともに一般消費者向けの Wi-Fi が登場したことで、Linux を搭載したデバイスは適応を迫られることになりました。IEEE 802.11 規格や Apple の AirPort 技術の登場を受け、HP の Linux 開発者である Jean Tourrihes は、Wireless Extensions（wext）を開発せざるを得なくなりました。これは、アプリケーションが無線ドライバーと通信できるようにするためのコマンド群です[26,27]。Wi-Fi が通信容量と機能を強化する中で、ハードウェア企業は自社の無線通信制御ロジックを独自技術として開発していました。その結果、業界は分断され、チップメーカーは製品ごとに Wi-Fi ロジックをゼロから書き直さなければならない状況に陥りました。やがてオープンソース開発者たちは、汎用的かつオープンソースの Wi-Fi 準拠フレームワークを、カーネルのコア部分に直接組み込みました[28]。このフレームワークの導入により、チップメーカーが自社の新しいチップをノート PC メーカーやクラウドベンダーに採用してもらうためには、Linux カーネルのオープンソース アーキテクチャに対応することが不可欠となりました。その結果、Linux は半導体業界全体に対し、単一の共通ソフトウェア標準に沿って協力するよう促すことになりました。これにより、次世代 Wi-Fi 技術の市場投入までの期間が劇的に短縮され、無線チップの製造コストが低下しました。さらには、ネットワーク機能の仮想化（NFV）、ソフトウェア定義ネットワーク（SDN）、クラウドネイティブ ネットワーキング、そしてハイパースケール データセンター アーキテクチャの普及が大きく加速することとなりました[28]。

Linux は、オープンソースのルーター開発や標準化の取り組みを通じて、世界の Wi-Fi インフラ市場を席巻するに至りました。2002 年、Linksys は開発コストを削減するために、修正を加えた Linux カーネルを搭載したルーター WRT54G を、そのソースコードと共に公開しました[29]。Software Freedom Law Center による GPL-2.0 の遵守徹底の動きは、カーネルのライセンスが法廷で強制力を持つことを明確にしただけでなく、Linux ベースのルーターをコミュニティによる改善や修正に向けて開放する道も開きました。これにより、現在広く普及している OpenWrt のようなプロジェクトが生まれ、高価な企業向け機器に匹敵する性能を備えた安価な家庭用ルーターが数多く登場することとなりました[29]。今日、Linux は家庭用 Wi-Fi ルーター、企業向けアクセスポイント、スマートホーム用 IoT ハブの基盤となるオペレーティングシステムとして機能しており、Linux ベースのテレビ用セットトップボックスなどとも接続されています[30,31,32,33]。例えば、Samsung のテレビはすべて、Linux をベースとしたオープンソースのソフトウェアスタックである Tizen OS で動作しています[34]。

コモディティハードウェアへの移行は、Linux カーネルを基盤とする次世代のオープンソース　ネットワーキング プロジェクトの基礎を築きました。DPDK（Data Plane Development Kit）のようなプロジェクトはパケット処理性能を最適化し、SONiC や DENT は Linux を拡張してスイッチ向けのオープンなネットワーク OS を実現しています。また、OPI（Open Programmable Infrastructure）プロジェクトは DPU や IPU のためのプログラマブルなインフラを標準化し、LF Networking プロジェクトは通信ネットワークやクラウドネットワーク全体にわたるオーケストレーション、自動化、相互運用性を統合する取り組みを行っています。これら一連のプロジェクトは、Linux がプログラマブルかつオープンなネットワーク インフラのための共通プラットフォームへと進化していく、自然な流れを体現しています。

Linux Foundation の Arpit Joshipura (General Manager of Networking, Edge and IoT) は、次のように振り返ります。「Linux は、ネットワーキングをプログラム可能かつオープンにし、誰にでも利用可能なものへと変えました。今日のネットワーキングプロジェクトは、その基盤の上に構築され、同じ原則をオペレーティングシステムの枠を超えて拡張することで、クラウド、エッジ、通信、AI インフラのための相互運用可能なプラットフォームを生み出しています。オープン コラボレーションは、イノベーションが業界全体での採用へと拡大していくことを促進し続けています。」

自動車

自動車業界における Linux の導入は、現代の自動車のあり方を再定義し、メーカーとユーザーの双方に多大な価値をもたらしました。これにより、自動車は単なる機械装置から、高度なモバイル データセンターへと変貌を遂げたのです。この進化を牽引しているのが、Linux Foundation が主導するオープンソース プロジェクト Automotive Grade Linux（AGL）です。AGL は、それまで断片化していたプロプライエタリー ソフトウェアシステムに取って代わる存在になりました。Linux カーネルがこの変革の核となるエンジンを提供する一方、Yocto Project の活用により、自動車メーカーは自社の特定のニーズに合わせた Linux ディストリビューションを構築できるようになりました。その成果として生まれたのが、AGL の Unified Code Base（UCB）です。これは、自動車に必要なソフトウェア スタックの約 80%をカバーする、強固な基盤となっています。さらに AGL は、自動車業界が求める長期サポートにも対応しており、15 年という車両のライフサイクル全体にわたって、車載ソフトウェアの安全性と保守性を維持することを可能にしています。

この統合された Linux ディストリビューションがもたらす実世界へのインパクトは、2018 年型 Toyota Camry の発売によって広く実証されました。同車は、インフォテインメント システムに AGL プラットフォームを採用した初の量産車となりました。この重要な節目は、各社が個別に進めていた研究開発から脱却する転換点となり、Mazda、Mercedes-Benz、Subaru、Suzuki、Panasonic をはじめとする、幅広いメンバーによる協調体制の構築へとつながりました。AGL の Senior Director of Community を務める Walt Miner は次のように述

Linux カーネル開発者シリーズ：Paul E. McKenney



Paul E. McKenney は 20 年以上にわたって Linux 開発に携わってきましたが、彼がコンピューターの世界に関わり始めたのは Linux の登場以前のことです。当時は、ハイエンド データベース サーバー向けのプロプライエタリーな並列 Unix システム DYNIX/ptx の開発に携わっていたほか、それ以前にはネットワーク研究にも従事していました。Paul が初めて Linux の存在を知ったのは 1990 年代半ば、Mac からデータを取り出す作業を手伝ってほしいと頼まれた時のことでした。そのすぐ後、Linux ネットワーク サブシステムのメンテナーを務めていた Alexey Kuznetsov から、Paul が 1990 年に発表した確率論的フェアスケジューラーに関する論文について問い合わせがありました。Kuznetsov はその論文のアイデアを Linux に実装し、カーネルコード内のコメントに Paul の名を引用しました。こうして、Paul にとっての Linux への最初のコントリビューションは、コードそのものではなく、参考文献の記載という形で行われることになりました。

当時、Paul は Sequent で DYNIX/ptx の開発に携わっていましたが、2000 年頃に IBM が Sequent を買収したことを機に、Linux 開発へと軸足を移しました。彼はさまざまな課題に取り組みましたが、自身が最初に提出したパッチがどのようなものだったのかは覚えていないといいます。「採用されたものもあれば、そうでないものもありました。本当に大変でしたよ」と言っています。彼は、Linux カーネル全体で広く利用されている、極めて重要かつ複雑な並行処理メカニズムである Read-Copy Update（RCU）のリード メンテナーとなりました。2010 年、Paul は RCU のメンテナー体制について 50 年計画を策定しました。これは、共同メンテナーを迎え入れ、自身の任期が終わった後もこのサブシステムが存続できるようにするためのものでした。彼によれば、こうしたメンテナンス作業には「成文化と指導が含まれており、RCU に関する用語や定義といったものはすべて、他の人々と議論するために自分たちで作り出さなければなりませんでした。30 年間、人々に説明し続けてきたことで、そのスキルも向上しました。」と語ります。

Paul はコードを書くだけでなく、オープンソースの教科書『Is Parallel Programming Hard, And, If So, What Can You Do About It?』を執筆し、rcutorture のようなテストスイートを構築したほか、Linux カーネルメモリモデル（LKMM）プロジェクトにも携わっており、世界中の開発者のメンターとしても積極的に活動しています。彼がなぜコミュニティでの活動を続けているのかについて、本人は次のように語っています。「理由の一つは、責任感です。Linux カーネルコミュニティは、RCU を受け入れた際、私に大きな信頼を寄せてくれたのです。」

Paul へのインタビュー全編は **My Life as a Linux Kernel Developer and Maintainer with Paul E. McKenney**[35] を、是非ご視聴下さい。

べています。「Toyota の AGL に対する取り組みは、2026 年に刷新された RAV4 において、オープンソースの組み込み Linux システム用 UI 開発環境 Flutter エンジンと最新の AGL コンポーネントを活用した新しい UI デザインが導入されたことで、さらに強化されました。」

共通プラットフォームにコントリビューションすることで、メーカーは、接続性やハードウェアの抽象化といった基本的な技術的課題を個別に解決する必要がなくなります。その代わり、一貫した環境を活用できるようになり、ソフトウェアを一度開発すれば、それを複数の車種に展開することが可能になります。こうした投資の共有によって、市場投入までの時間が短縮し、各社がリソースを集中させて自社ブランドならではの独自のデジタル体験を顧客に提供することに注力できるようになります。

現在、AGL は SoDeV リファレンスプラットフォームを通じて、ソフトウェア定義車両（SDV）への移行を推進しています。これは、AGL Linux ディストリビューションを独立した仮想マシン上で実行するためのフレームワークです。このアーキテクチャにより、単一の高性能コンピューター上で、安全性が極めて重要な機能と、インストルメント クラスター（計器類）やエンターテインメント アプリケーションを、互いに干渉させることなく同一ハードウェア上で管理することが可能になります。これにより、たとえコンシューマー向けアプリケーションでエラーが発生した場合でも、デジタル ダッシュボードの信頼性を維持できます。業界の調査によると、こうしたオープンな Linux ベースのアーキテクチャを採用することで、車両の研究開発コストを最大 40%削減できる可能性があります。また、Linux Foundation の報告によれば、現在 150 以上の組織が、モビリティの次なる時代を切り拓くべく、この実用可能な標準規格のもとに結集しています[36,37]。



2025 年 12 月、東京で開催された *Automotive Linux Summit* にて登壇する *Dan Cauchy*

映画およびゲーム

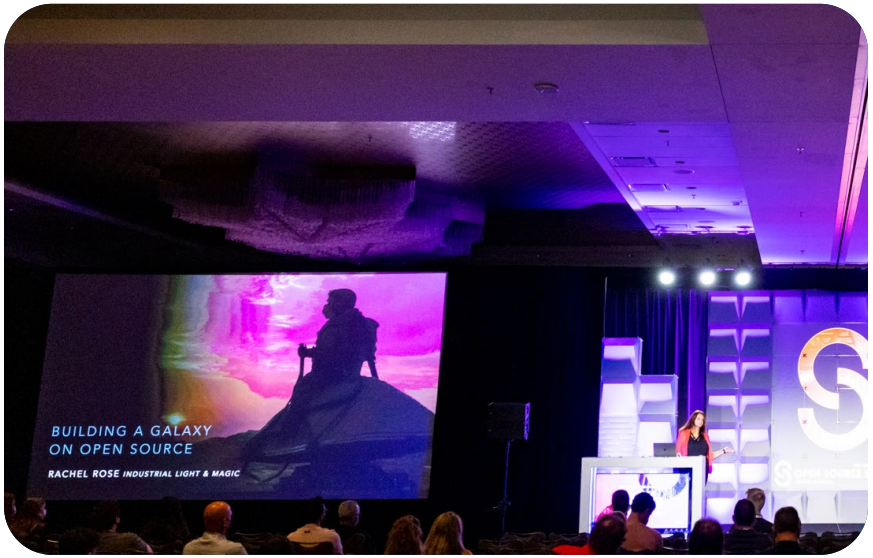
エンターテインメント分野において、Linux カーネルは、映画やゲーム制作における現代のデジタル　パイプラインを支える基盤エンジンであり、業界内におけるオープンソース コラボレーションを促進する触媒としての役割を果たしています。1990 年代のクリエイティブ ブームにおいて、VFX（視覚効果）や CG アニメーションの制作スタジオは、デジタル ストーリーテリングの可能性を押し広げるために、Silicon Graphics（SGI）が提供する、高価なプロプライエタリー Unix ベースのワークステーション IRIX に大きく依存していました。しかし、SGI がハリウッドの映画制作者たちが求める高度かつ急速に変化する技術的ニーズから戦略の軸足を離したことを受け、2000 年 6 月、業界全体を巻き込んだ歴史的な連携が結成されました。各スタジオは、自社の制作パイプラインをオープンソースのオペレーティングシステムである Linux へと移行させることに共同で合意しました[38]。

この移行の初期段階は、試行錯誤と、極めて高度なエンジニアリングの創意工夫によって達成されたものです。James Cameron 監督の映画『タイタニック (1997)』は、デジタルエフェクトの大部分（特に複雑な水のシーン）を、Red Hat Linux で稼働する大規模なレンダーファーム（コンピュータークラスター）を用いて制作した最初の主要な長編映画となりました。そのすぐ後、DreamWorks Animation の映画、『シュレック (2001)』が、全編コンピューターアニメーションによる長編映画として初めて、完全に Linux 環境下で制作されました。同作では 1,000 台以上の Linux サーバーが活用され、大容量で複雑なデータ処理や毛髪の表現処理が行われました[38]。このシフトにより、各スタジオは高価なハードウェアへの依存から脱却し、汎用的なインテル製プロセッサを用いることで、技術者たちが使い捨て可能なコンピューティングと呼ぶ手法を実現しました。各スタジオは、その市場における影響力を結集させることで、主要な商用ソフトウェアベンダーに対し、Alias Wavefront の Maya、SideFX の Houdini、Pixar の RenderMan といった主要な制作アプリケーションを、Linux 環境へ直接移植するよう強く促し、これを実現させました。

> **今日、Linux はハリウッドの制作現場を完全に席巻しており、主要な VFX（視覚効果）、アニメーション、デジタルレンダリングのクラスターの事実上 100% が Linux のノード環境上で稼働しています[39]。**

Academy of Motion Picture Arts and Sciences（映画芸術科学アカデミー）は、Linux に関する業界全体のこうした協力体制を基盤として、2018 年 8 月に Linux Foundation と提携し、Academy Software Foundation（ASWF）を設立しました。これにより、Linux において実践されてきた協調的なアプローチが、映画スタジオで使用されるソフトウェアにも導入されることとなりました。

今年、Linux が業界に影響を及ぼすための新たな機会が生まれました。Valve は、2018 年に初めてリリースした Windows 互換レイヤー Proton をさらに発展させ、Proton 11.0 を導入することで、ハードウェアからの独立性を大きく前進させました。このバージョンには Wine 11 が統合されているほか、ntsync と呼ばれる全く新しいカーネルレベルの同期フレームワークが採用されており、Linux 上で動作するマルチスレッド対応の Windows ゲームにおいて CPU オーバーヘッドを劇的に削減します。さらに画期的なのは、ARM64 との互換性を実現するために FEX-Emu が統合されたことです。これにより、本来 x86 アーキテクチャ向けに構築されたゲームを、Linux が動作する ARM64 ベースのモバイルハードウェア上でネイティブに実行できるようになりました。これは、Valve のスタンドアローン型 VR ヘッドセットや、Snapdragon を搭載した Steam Deck のようなデバイスの基盤となるものであり、オープンソースの Linux 環境において、エネルギー効率に優れた携帯型デバイスで複雑な PC 向けタイトルをプレイすることを可能にします[40]。



2022 年 6 月、*Open Source Summit North America* に登壇している *Rachel Rose*

ヘルスケア

ヘルスケア分野は、Linux の歴史において長らく矛盾を抱える領域であり続けてきました。データ集約型かつ高度な計算処理を要する業務が中心であるにもかかわらず、オープンソースを本格的かつ大規模に導入する動きは、他の産業と比べても大きく遅れていました。他の業界とは異なり、ヘルスケア分野

における Linux カーネルの存在感は、IT 部門での利用という形ではなく、医療用画像診断装置、検査機器、診断用ワークステーションといった組み込みシステムにおいて顕著でした。ソフトウェアの不具合が人命に関わる事態を招きかねないというヘルスケア分野の性質上、金融や通信といった他業界と比べると、Linux カーネルの導入は比較的ゆっくりと進んできました。その一方で、Linux は、安定性が高く、広く精査されており、コスト効率にも優れています。さらに、単一のベンダーが独自に行うよりも迅速に脆弱性を特定し、修正パッチを適用できる世界規模の開発者コミュニティに支えられているという点は、ヘルスケア業界にとって Linux が極めて合理的な選択肢であることを示しています[41]。

ヘルスケア分野における Linux の導入は、一朝一夕に成し遂げられたものではありません。医療機器の相互接続性が高まるにつれ、メーカー各社は、マルチタスク処理、グラフィックス、ネットワークといったワークフローを扱えるプラットフォームを必要とするようになりました。こうしたニーズに応えたのが Embedded Linux です。これにより、特に医療機器メーカーは、プロプライエタリー技術で閉ざされた開発環境（サイロと呼ぶことがある）を解消し、ハードウェアや機器の市場投入までの期間を短縮できるようになりました。

ヘルスケア業界では、最先端のイノベーションを可能にしつつ、厳格な安全要件を満たすコンピューティング インフラが求められていました。こうしたニーズに応える形で立ち上げられたのが、**Enabling Linux in Safety Applications（ELISA）**プロジェクトです。2019 年 2 月に Linux Foundation によって開始された ELISA は、参加メンバーが協力して共通のツールやプロセスを構築し、企業が認証取得に必要な安全要件を満たせるよう支援しています。同プロジェクト内には Medical Devices Working Group が設置されており、特に医療機器ソフトウェアのライフサイクルを規定する国際規格 IEC 62304 に焦点を当てて活動しています。この規格は、米国食品医薬品局（FDA）が、医療機器の規制において、市販前申請を裏付けるために用いる合意規格（コンセンサス スタンダード）として認めているものです[42]。

デバイス層を超えて、Linux は複雑な計算を伴う医学研究の推進役となっています。その顕著な例として、National Institutes of Health（NIH）が挙げられます。同研究所では、データシーケンシング（塩基配列解析）プロジェクトの処理に不可欠な、Linux ベースの HPC（ハイパフォーマンス コンピューティング）クラスターが運用されています。ヒトゲノム計画に端を発する大規模なゲノム解析パイプラインや、広く利用されている FMRIB Software Library や FreeSurfer といった神経画像解析スイートは、いずれも Linux 環境を前提としており、脳スキャン画像や遺伝子マップのような膨大なデータセットの処理を支えています[43,44,45]。こうした高度な計算能力への要求と、充実したツール群のエコシステムが組み合わさったことで、Linux カーネルは医療研究の分野に特に適したものになっています。

今日、Linux カーネルは、ヘルスケア業界のあり方を大きく規定する最新の運用制御技術と診断技術において、中核となる処理エンジンとしての役割を担っています。その影響は、高解像度の医療画像処理システムや複雑な臨床検査分析装置から、患者用ウェアラブルモニターや薬剤投与ポンプに組み込まれた小型システムに至るまで、幅広く及んでいます。Linux は、ヘルスケア機器を接続し、正常に機能させるためのコンピューティング層を提供しています。さらに、ELISA コミュニティの協力により、これらの機器は安全性の認証を取得することも可能となっています。



2024 年 4 月にシアトルで開催された *Embedded Open Source Summit* で講演を行う *Kate Stewart* と *Tim Bird*

製造業と市民社会のインフラ

製造業や社会インフラの分野において、Linux カーネルの安全性と安定性は極めて重要な特性です。Linux は、工場現場におけるプログラマブル ロジック コントローラ（PLC）、監視制御・データ収集（SCADA）システム、エッジ コンピューティング ゲートウェイ、スマートセンサーといった機器と、企業ネットワークとを橋渡しする重要な処理層を担っています。物理的な製造インフラは、設備が最大 40 年間にわたって安全かつ継続的に稼働することが求められるなど、極めて長いライフサイクルを有しており、一般的なソフトウェアのサポート期間とは技術的に大きく乖離しています。こうした産業界のニーズに応えるべく立ち上げられたのが CIP(Civil Infrastructure Platform）プロジェクトです。同プロジェクトは、システムの安定性を損なうことなく、また独自仕様による断片化を招くこともなく、数十年にわたって Linux カーネルを維持するための、信頼できる協調体制の枠組みを構築しました。

長寿命化のギャップを解消するため、CIP プロジェクトは特定の長期サポート（LTS: Long-Term Support）Linux カーネルを選定し、それを超長期サポート（SLTS: Super Long-Term Support）カーネルへと進化させています。標準的な LTS カーネルのサポート期間は数年程度ですが、CIP は SLTS リリースに対して 10 年から 20 年という極めて長い期間、セキュリティ確保とサポートを提供することを約束しています。Siemens、Toshiba、Renesas、Texas Instruments、Cybertrust といった業界各社がこの枠組みの運営に積極的に参画しており、各社の製品用オペレーティングシステムの中核エンジンとして SLTS を組み込んでいます。この共通の基盤を活用することで、企業は多様な製品ラインにわたって、極めて安定した単一のソフトウェア基盤を展開できるようになり、数十年にわたる保守・運用を簡素化することが可能となります。

数十年にわたるサポートに加え、こうした物理環境ではマイクロ秒単位の決定性が求められます。このニーズに応えるのが Real-Time Linux (RT-Linux) プロジェクトです。コミュニティによる長年の取り組みを経て、RT-Linux の核となる PREEMPT_RT パッチセットが、Linux 6.12 において正式にメインラインへ統合されました。主要メンバー組織である CIP は、RT-Linux の機能を自身の SLTS のベースラインに取り込んでおり、これにより、厳密な応答時間が求められるインフラ機器において、ハードリアルタイムの予測可能性と、企業の大規模な運用環境に耐えうる極めて高い安定性を両立させています。

現実世界の物理的ユースケースにおいて求められる信頼性とリアルタイム応答性は、KernelCI プロジェクトのフレームワーク上に構築された自動テストパイプラインによって実現されています。Linaro が主導して開始し、後に BayLibre や Collabora が参画したこのフレームワークは、カーネルテストのインフラに特化した独立した基盤へと発展しました。都市や産業の重要なインフラにおける障害は、許容しがたい物理的リスクを招く恐れがあるため、CIP プロジェクトの Testing Work Group では、KernelCI を活用して、実際の産業用リファレンスハードウェア上で SLTS カーネルの継続的な検証を行っています。この自動テスト環境により、バックポートされたセキュリティパッチやバグ修正が、リグレッション（機能後退）を引き起こしたり、レガシーシステムを損なったりしないことが保証されます。このプロセスは非常に厳格なものですが、不可欠な手順であり、IEC 62443 や EU Cyber Resilience Act のような厳格なグローバル セキュリティ基準への準拠を可能にする道筋を切り拓いてきました[46]。

CIP および KernelCI の各コミュニティの下で、社会インフラや製造業の企業は、高度な技術的原則を進化させてきました。これらの活動においては、「upstream-first（上流優先）」の原則に基づき、極めて効率的かつ好循環を生むコントリビューションのサイクルへと直接つながっています。CIP の開発者はまず、バグ修正や技術的な機能強化をすべて、Linux カーネルのメインラインのコードベースに対して直接提出します。それらの変更が、より広範なオープンソース コミュニティによって検証され、マージされた後に初めて、SLTS リリースへとバックポートされます。このアプローチは、独自の改変による断片化を防ぎ、参加企業にとっての総所有コストを大幅に削減するだけでなく、相互に利益をもたらす好循環を創出しています。

CIP の専門ワーキンググループは、テスト、検証、そしてアップストリームへのコントリビューションに積極的に取り組むことで、安定カーネルや Debian のベースラインを改善しています。これにより、製造業や社会インフラの分野において、より広範な Linux エコシステムの安全性とレジリエンスが高められています[47]。これらのコミュニティは、Linux Foundation の傘下にあるオープンソースプロジェクトの中でも、特に長い歴史を持つものの一つです。KernelCI プロジェクトは 2014 年に、CIP プロジェクトは 2016 年に開始されました。これらは一つの傾向を物語っています。すなわち、Linux カーネルが普及・拡大するにつれて、業界に特化した、より専門的なオープンソース コラボレーションの形態もまた発展していくということです。

> **Linux カーネルが普及・拡大するにつれて、業界に特化した、より専門的なオープンソース コラボレーションの形態もまた発展していきました。**

宇宙、ドローン、自律型システム

航空宇宙および自律飛行の分野において、Linux カーネルは独自の足跡を刻んできました。数十年にわたり、航空宇宙システムのアーキテクチャは、極めて制約の厳しいリアルタイム オペレーティングシステム（RTOS）を搭載した、高価で耐放射線性能を備えたマイクロコントローラーにほぼ全面的に依存していました。しかし、前述したヘルスケア分野の事例と同様に、現代的な宇宙ミッションの登場により、従来のハードウェアの限界をはるかに超えるコンピューティング インフラが求められるようになりました。宇宙分野では、リアルタイム コンピューター ビジョン、自律着陸のための計算処理、そして高スループットなデータ処理が必要とされており、Linux カーネルはこれに理想的なものでした。Embedded Linux は、マルチコア プロセッサ上で複雑なアルゴリズムを実行できる、オープンかつ極めて適応性の高いプラットフォームを提供することができました。これにより、宇宙関連機関・組織や民間航空宇宙のパイオニアたちは、基本的なテレメトリから、地球近傍および深宇宙環境における完全自律型の運用に至るまで、Linux に移行することが可能となりました。

この進化における画期的な出来事として、NASA の火星ヘリコプターが挙げられます。同機は 2021 年、地球以外の惑星において、初となる動力制御飛行を成功させ、歴史に名を刻みました[48]。火星ヘリコプターは、Embedded Linux を搭載した市販のプロセッサをメインの機上コンピューターとして稼働させ、さらに、NASA Jet Propulsion Laboratory (JPL) が開発したオープンソースのフライトソフトウェア フレームワーク F Prime を採用することで、オープンソースソフトウェアが極限の地球外環境においても機能し、優れた性能を発揮できることを実証しました[49]。地球との間に約 15 分もの通信遅延がある中で、火星ヘリコプターは Linux を駆使して、火星の希薄な大気中で視覚慣性航法アルゴリズムを実行し、センサーデータやカメラ映像をリアルタイムで解析することで、機体の安定性を維持しました[50]。この成功は新たなパラダイムを切り拓くものとなり、Linux を搭載した低コストのハードウェアであっても、過酷かつ遠隔の環境下で、ミッション遂行に不可欠な高い負荷を伴う処理を担えることを証明しました。

惑星探査にとどまらず、Linux は、地球低軌道上のコンステレーション ネットワーク、貨物輸送機、そして各種衛星システムを支える基盤となっています。国際宇宙ステーション（ISS）では、重要な実験の最中にシステムがクラッシュする事態を避け、運用上の信頼性を確保するために、主要なノート PC や指令システムで Linux ディストリビューションが稼働しています[51]。同様に、民間宇宙飛行の主要企業各社も、宇宙航空機向け電子機器、ロケットブースターの回収工程、衛星インターネット コンステレーションといった分野で Linux を活用しています[52]。中でも Starlink は、3 万 2000 基もの高度に相互接続された衛星で構成されており、宇宙空間における Linux の最大の導入事例となっています[53]。宇宙産業におけるイノベーションをさらに加速させるべく、現在 **Space Grade Linux** という新たなインキュベーション プロジェクトが進められています。これは、さまざまな宇宙関連の用途で共通して利用できる Linux の標準化された基盤層を構築し、業界全体が足並みを揃えて前進できるようにすることを目指しています[54]。



出典：*Andrea Luck の Ingenuity - The First Epic Mars Helicopter。Nasa's Ingenuity - Sol 44 より。CC BY 2.0。*
https://commons.wikimedia.org/w/index.php?curid=145282930

この移行は、衛星の軌道上で始まったわけではありません。20 年もの間、同様の機能は地上に近い場所で成熟を続けてきました。そこでは、Linux が現代のドローンシステムにおける標準的な構成要素となっています。一般的な自律型ドローンは、リアルタイムのフライトコントローラーと、Linux を搭載したコンパニオンコンピューターを組み合わせる構成をとります。このコンパニオンコンピューターは、コンピュータービジョンや障害物回避から、通信接続、機体群管理に至るまで、より高度な自律機能を担います。その際、オープンソースのミドルウェアである **Robot Operating System 2**（ROS 2）が活用されるケースも増えています。こうしたアーキテクチャは、PX4 オートパイロットやオープンハードウェア規格 Pixhawk を管理する Linux Foundation のプロジェクト **Dronecode Foundation** により、オープンな環境で標準化されています。自律型エアタクシーもこの設計をさらに発展させたものであり、LiDAR、レーダー、光学カメラといった複数のセンサーから得られる情報を統合（センサーフュージョン）するために Linux が利用されています。シミュレーションから継続的インテグレーション、飛行ログの分析に至るまで、ドローン開発のライフサイクル全体が、事実上すべて Linux 上で実行されています。

自律飛行は航空当局の厳格な監督下にあり、安全性が極めて重要視される業界分野であるため、航空宇宙エンジニアは、前述の ELISA プロジェクトのようなオープンソースの取り組みを通じて協調しています。Boeing は 2022 年に Premier メンバーとして ELISA に参加し、Aerospace Working Group の立ち上げを支援しました。Linux はすでに Design Assurance Level D の認証を受けたシステムで使用されており、メインラインカーネルの PREEMPT_RT は決定論的な振る舞いが求められる飛行制御アプリケーションの要件を満たしており、コミュニティはより重要な役割のためのアシュアランス事例を体系的に構築しています。認証ツールさえもオープンに開発されています。University of Illinois と Boeing が共同開発した構造的カバレッジツールは、カーネルを DO-178C（民間航空機に搭載するソフトウェアの安全性認証基準）の要件を満たしているかを測定することができ、2025 Digital Avionics Systems Conference で最優秀論文賞を受賞しました。

「Linux はドローン分野において、着実に、その実力によって、そして一つひとつの重要な決断を積み重ねることで勝利を収めました。宇宙における自律システムにも、同様の規律と信頼が求められます。オープンな環境で構築された共通の基盤こそが、地球上から星々に至るまで、自律技術を創出し、拡大していくために不可欠な信頼を築くのです」と、Dronecode Foundation の Executive Director である Ramón Roche は述べています。

Linux カーネル開発者シリーズ：Julia Lawall



Julia Lawall のソフトウェア開発者としての経歴は、プログラミング言語およびプログラム変換に関する学術研究に端を発しています。1990 年代初頭に博士号を取得し、長年にわたってプログラミング言語の学術研究に携わった後、2000 年頃からオペレーティングシステムやカーネルの研究へと軸足を移しました。2004 年、Linux 2.4 から 2.6 への移行期に、彼女と研究チームは 1 年以上の歳月をかけて、カーネル API に加えられた数百もの変更を一つひとつ手作業で分析しました。その過程で、インターフェースに変更が加えられると、すべてのデバイスドライバーに対して手作業で修正を反映させる必要があり、しかもそのリファクタリング作業にはミスが起こりやすいという問題が明らかになりました。そこで彼らは、カーネルのバージョンをまたいでデバイスドライバーの API 更新を自動化することに取り組み始めました。

Julia と彼女の同僚である Gilles Muller、Yoann Padioleau、René Rydhof Hansen は、Coccinelle というセマンティック パッチ適用ツールを開発しました。これは、開発者がパッチ構文を用いて C 言語コードの構造的な変換を容易に記述できるように設計されたものです。彼らは当初、比較的目立たないエラーを修正する小さなパッチをいくつか送っていましたが、やがてサブシステムをまたぐ 100 以上の修正を含む大規模なパッチを提出するに至りました。このパッチに対し、メンテナーの Andrew Morton から、複数の修正を一つにまとめることについて異論が唱えられました。当時、MAINTAINERS ファイルには、どの修正をどのメンテナーに送るべきかを容易に判断するための十分な情報が記載されていなかったためです。Morton はひとまずそのパッチを受け入れることに同意しました。最終的に Coccinelle は Linux カーネル開発における不可欠なツールとなり、現在までに 1 万件以上のカーネルコミットで言及されています。Julia は次のように述べています。「Coccinelle がうまく機能する理由は、Linux カーネルが非常に多様でありながらも、多くの共通点を持っているからです。また、共通の基準が適用される、非常に強固なレビュー文化も存在します。コードの構造や主要な API の形が共通しているのです。――まさに、Coccinelle のようなツールにとって理想的な環境なのです。」

現在、Julia はパリの Inria でシニアリサーチャーを務める傍ら、Coccinelle のメンテナーとしても活動しており、Linux Foundation の Technical Advisory Board のメンバーも務めています。また、Outreachy や LFX Mentorship といったプログラムでメンターを務めた経験もあり、現在はスケジューラーのパフォーマンスやソフトウェアの形式的検証に関する研究に注力しています。なぜコミュニティに留まり続けているのかという問いに対し、Julia は次のように語りました。「ここは本当に素晴らしいコミュニティです。私たちの目の前には、非常に多様で興味深い、非常に素晴らしく、役に立つリソース――膨大なコードベース――が存在しています。そのどの部分に深く入り込んでも、そこには全く新しい刺激的な世界が広がっていて、各要素がどのように組み合わさっているのかを解き明かしていく面白さがあります。」

Julia のインタビュー全編は、**My Life as a Linux Kernel Developer and Maintainer with Julia Lawall** を、是非ご視聴下さい [55]。

Linux の技術的なインパクト

Linux の歴史は、既存のプロプライエタリー テクノロジーの品質と有用性を大幅に凌駕し、それをオープンソース化し、それによって最先端ソリューションへのアクセスを民主化した歴史です。

> **その誕生以来、Linux は、その時々の技術的要件に応じる形で拡張、進化してきましたが、同時に新たな運用構造、ソフトウェア、ユースケースに影響を与え、またそれらから影響を受けてきました。**

以下においては、Linux がどのように直面するコンピューティング ニーズに適合するように進化してきたか、またそれらのコンピューティング エコシステムがどのように Linux カーネルを取り入れ、それを中心に進化してきたかを説明します。

個人および企業向けコンピューティング

企業は、データ処理やネットワークに Linux を使用することにより、安定性、カスタマイズ性、拡張性を享受しながらコストを削減できることをすばやく学びました。Red Hat の創業者たちは、Linux のカスタマイズ性を企業向けの主要な差別化要因と位置づけ、「より優れている、より速い、より多くの機能を備えている」からではなく、適応性というただ一つの特徴を売り文句に、Unix の代替として Linux の販売を開始しました[56]。1990 年代後半、IBM の開発者たちは Linux を使った実験を開始し、S/390 メインフレーム上で動作させるためのパッチを公開しました。[57] 1999 年末の Linux 2.2.13 のリリースにより、Linux カーネルはエンタープライズ ワークロードや大規模 I/O スループットの処理においてメインフレームと同等レベルに対応できるようになりました。今日では、現代風ワークロードの大部分が Linux 上で稼働しています。 [58]メインフレームに真似のできないこととして、企業は安価な PC 上で Linux を（Apache、MySQL、Python といったプログラミング言語と組み合わせて）実行することにより、インターネット ビジネスをスケールアップするための世界クラスのサーバーを提供できることを認識し始めました。コンシューマー インターネットの時代において、Linux はサーバー室でますます魅力的な存在となりました。[59]今日、Linux は Google、Amazon、Netflix、Uber、CERN、主要な証券取引所などのサーバーで稼働しています。Fortune 500 の 90% 以上が Red Hat ソリューションを使用しており、[60] Linux はあらゆる Web サイトにおいて最も広く使われているオペレーティングシステムとなっています。[61]

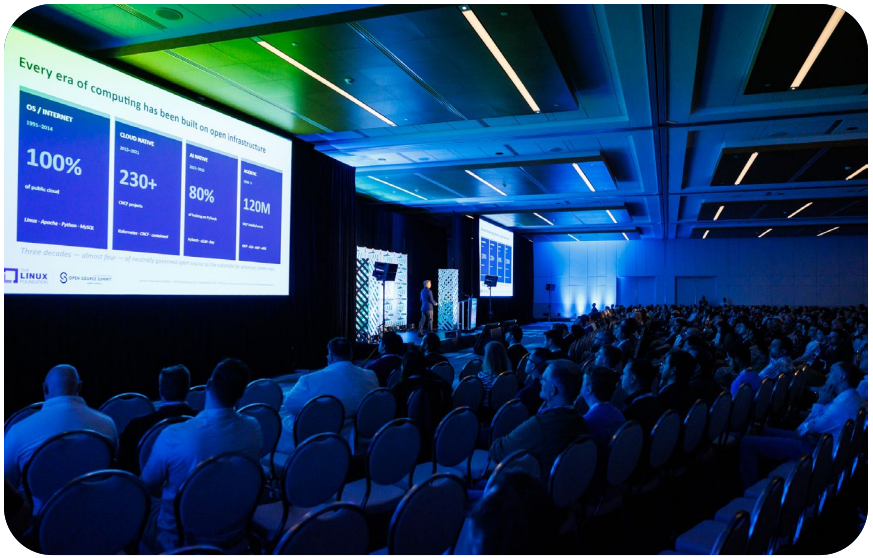
Linux はまた、標準 PC 上で高いパフォーマンスの環境を提供することにより、ワークステーション市場を破壊しました。高価なワークステーション マシンを購入する代わりに、PC を購入して Linux をインストールすれば、同じレベルのパフォーマンスを得ることができます。これに注目し始めたのが、高度なコンピューティング能力を必要とする業界でした。自動車業界や映画業界はその先駆者であり、DreamWorks、Pixar、Volkswagen などの企業が 2000 年代初頭に Linux ワークステーションを導入しました。[62,63,64] また、国防、政府機関、金融サービスといった厳格に規制が適用される分野においても、Linux はワークステーション用システムとして選好されています。その理由は、Red Hat のワークステーション向けライセンスが提供するセキュリティ、安定性、そして長期サポート保証にあります。[65]

ノート PC や携帯電話の使用増加にともなってデスクトップの利用形態がモバイル化するにつれ、Linux は省電力化が求められるようになりました。もともとデータセンターでの運用を想定して設計されていた Linux カーネルには、ハードウェアタイマーが組み込まれており、これが CPU を強制的にウェイクアップさせて、実行すべきタスクの有無を 1 秒間に数百回から数千回も確認させていました。 2007 年、モバイル機器のバッテリー持続時間向上に対する需要が高まる中、開発者たちはより効率的な電力管理の実現に向けて設計を見直し、Tickless Kernel を開発しました。この仕組みでは、カーネルがイベントの有無を確認し、次のイベントが発生するタイミングに合わせてタイマーが作動するように設定されます。[66]

モバイル コンピューティングの普及により、世界における Linux の利用は大きく拡大しました。一見すると、デスクトップ PC やノート PC 向け OS としての利用率は約 3% にとどまり、ごくわずかなもののように思えるかもしれません。[67] しかし、大きな販売数を誇る Chromebook が実際には Linux を遥かに普及させています。ブラウザベースの ChromeOS は根本的に Linux ディストリビューションであり、Chromebook を使用している人々は Linux を使用していることになります。[68] ノート PC 以外に、スマートフォンも個人・仕事向けコンピューティングにおける Linux の市場シェアに寄与しており、Android は全世界でモバイル OS 市場シェアの 72% を占め、40 億台以上の Android デバイスが使用されています。[69]

Linux がコンピューティングにもたらした影響は、そのコスト効率とモジュラーアーキテクチャ以上の大きなものもあります。カーネルを中心としたオープンソース コミュニティの存在はメインフレーム コンピューティングにおける戦略的なビジネス上の決断の要因となり、その意義は今日まで受け継がれています。IBM の Linux イニシアチブを率いた元エグゼクティブ Irving Wladawsky-Berger が説明したように、「2000 年当時、IBM は Linux カーネルを理由として Linux を採用したわけではありません。当時のカーネルは、大幅な改善が必要であることを私たちは理解していたからです。しかし、私たちは Linux コミュニティに目を向け、そのあり方に非常に好感を抱きました。そして、IBM のような企業が開発にコントリビューションすれば、Linux は時とともに大きく進化していくだろうという確信を得たのです。実際に私たちはそれを実行しました。」[70]

カーネル メンテナーの Sasha Levin がインタビューで説明したように、Linux がサーバーおよびデータセンターの世界でオペレーティングシステムのデファクト スタンダードとなるにつれ、ますます多くの企業がアップストリーム コミュニティと足並みを揃え始めました。「コミュニティの良き市民になる以外に選択肢はありません。Linux の公式ソースツリー外のドライバーを持つことは苦痛でした…カーネルの進歩を追いかけることが苦痛でした。テクノロジーに関する苦痛であるだけでなく、経済的にも苦痛でした――Linux を使用するあなたの顧客が存在し、彼らのカーネル上で [あなたの] 製品を使用できることを期待しているのです。」



2026 年 5 月、ミネアポリスで開催された *Open Source Summit North America* で講演する *Jim Zemlin*

Linux カーネル開発者シリーズ：Shuah Khan



Shuah は、独立したコントリビューターとして最初のアップストリーム向けオープンソース コントリビューションを行う前は、長年にわたって Linux ユーザーでした。カーネル開発への彼女の参入の機会は、Android git リポジトリのコードがメインライン カーネルに持ち込まれた Android メインライン化の取り組みの最中に訪れました。彼女の最初のコントリビューションは Android staging ドライバーの削除に関連しており、このパッチは Greg Kroah-Hartman によって受け容れられました。彼女は自身のノート PC を使用して、LED ドライバーフレームワーク、いろいろなトリガーがどのように作用するか、およびドライバーの挙動をどのようにテストするかを迅速に学びながら、Android 機能の強化に取り組み続けました。「オープンソースで働くには異なるマインドセットを持たなければなりません」と彼女は説明しました。新しいコントリビューターに対して、彼女は「自分のコードを送り出し、多くの人々にレビューしてもらう勇気を持たなければなりません…自分よりもカーネルを良く知っている他の人々からのインプットを求めるのです…自分が実装する最初のアイディアに固執しすぎないでください」と説明しました。

Shuah のオープンソースでの体験は、さまざまな企業およびオープンソース活動における役割を通じて拡大しました。彼女は Hewlett-Packard で ProLiant ワークステーションに取り組み、そのさいに Linux ディストリビューション Red Hat および SUSE をサポートして、ファームウェアとカーネルの間の複雑な相互作用をデバッグし、また 2TB メモリ構成や IOMMU をサポートするためのバックポートを特定することなどに携わりました。その後、Samsung では、彼女はオープンソースグループに採用された最初のエンジニアの 1 人となり、そこではプロダクトチームのニーズとアップストリーム カーネル機能を橋渡しする役割を果たし、複数のカーネルリリースにわたって開発ボードを保守し、社内メンタリングクラスを主導しました。Linux カーネル コミュニティにおいて、Shuah は安定リリースのサポートを開始し、Linux Kernel Selftests のメンテナーシップを引き継ぎ、散在していたテストスイートをメインライン カーネルの Makefile に接続された一貫したフレームワーク駆動型のテストスイートへと変えしました。彼女はコミュニティのマインドセットをシフトさせて、テストスイート作成を機能開発のデフォルト部分へと有機的に組み込みました。また、メンターおよびメンティーとしてのバックグラウンドを活かし、試行としてメンティー 2 人から開始した Linux Foundation Kernel Mentorship Program を、自らが選んだ現実世界の課題に取り組む、毎年約 100 人以上の開発者をトレーニングするグローバルなイニシアチブへと築き上げ、成長させました。メンターシップ プログラムの成功を評価する際、彼女は「成果物を見てはいません…代わりに、私は彼らが夢をかなえることができる活力を授けているのです…それが何であろうと、彼らの道を見つけるために [メンティーに] 活力を与える乗り物なのです」と語りました。

Shuah のインタビュー全編は、**My Life as a Linux Kernel Developer and Maintainer with Shuah Khan** [71] を、是非ご視聴下さい。

クラウド コンピューティングとコンテナ

物理サーバーから仮想サーバーへの進化は、Linux カーネルとますます密接に関わり合いながら進展しました。物理サーバー上でアプリケーションを運用する際に生じるリソースの非効率性、スケーラビリティの課題、そして物理的な設置スペースの必要性といった問題が、仮想化技術の本格的な開発と普及を後押しすることとなりました[72]。当初、開発者は基盤となるハードウェアをエミュレートする仮想マシンを利用していましたが、その後、ハードウェアではなくオペレーティングシステムを仮想化するコンテナが開発されました。コンテナは、アプリケーションのコードを、必要なライブラリや依存関係すべてとセットにしてパッケージ化するものです。[72]

Linux は、仮想化とコンテナを効果的に展開するために時間とともに進化しました。2014 年末時点で、Linux カーネルは、コンテナをサポートするための 3 つの重要な機能を含んでいます：コンテナ内で実行される各アプリケーションがアクセスできるものを制限し、隔離されたネットワークスタック、IP アドレス、ファイルツリーを提供する Namespaces（ネームスペース）；アプリケーションが消費できる CPU、メモリ、ディスク I/O、およびプロセスの量を監視・管理する Control Groups（cgroups）；そして下層のコンテナイメージを維持しながらシステムの最上層のみを変更して、アプリケーションには統一されたものに見せるファイルシステム機能 OverlayFS。[73] Namespaces は長い時間（2002 年～ 2013 年）をかけてカーネルに導入され、Control Groups は 2008 年にメインカーネルにマージされ、OverlayFS は 2014 年末にマージされました。[74,75,76,77] Linux3.18 のもとでリリースされたこの最後のコントリビューションは、カーネルが自己完結したネイティブコンテナ プラットフォームとなったことを意味しました。そのリリース以前は、エンタープライズ プラットフォームの構築や Docker の実行は、カーネルに手動でパッチを当てるか、遅いストレージ ドライバーを使用することを意味していました。今日、パブリッククラウド インフラの約 90% が Linux 上で実行されており、世界中のクラウド ワークロードの 49.2% を動かしています。[78,79]

仮想化市場における Linux の圧倒的な優位性は、転じて、クラウドコミュニティが Linux を中心として展開する結果をもたらしました。何十億ドル規模の企業やソリューションが Linux ベースのコンテナをサポートするために次々と登場しています。たとえば、Docker プラットフォームは、人間が理解しやすいコマンドを Linux カーネルが実行可能な低レベルの要求へと変換することで、コンテナ作成の複雑さを簡素化します。また、カーネルレベルでパケット フィルタリングの観測を可能にし、同時に複雑さを削減しセキュリティを向上させる eBPF など、クラウド コンピューティングのさまざまな側面をサポートするプロジェクトも立ち上がりました[80]。さらに、クラウドからメインフレームに至るまで Linux が広く普及していることを背景に、Open Mainframe プロジェクトはハイブリッド クラウドのアプローチを可能にし、Linux OS、標準化されたコンテナ化、および共通の DevOps パイプラインを軸として両方のコンピューティング環境を連携させています。[81]

「クラウドネイティブという潮流は、Linux カーネルに直接組み込まれた中核的なプリミティブ――とりわけ cgroups、namespaces、eBPF――の存在があってこそ実現したものです。業界で『コンテナ革命』と呼ばれているものは、突き詰めれば、Linux カーネルが本来備えている分離機能やスケジューリング機能を基盤として、Kubernetes や Docker といったオーケストレーション フレームワークが構築されたことによるものです。35 年間を経て、Linux は現代のソフトウェア ワークロードの不可欠なランタイム環境であり続けています。」―― Chris Aniszczyk（CNCF CTO）

「リアルタイム」完成への 20 年：PREEMPT_RT

2024 年 9 月、ウィーンでのカンファレンスで、Thomas Gleixner は Linus Torvalds に印刷されたプルリクエストを手渡しました。[82] Linus はそれをマージし、Linux 6.12 はリアルタイムサポートを完備した最初のメインライン カーネルとなりました。その瞬間に、Linux を真のリアルタイム オペレーティングシステムにするための 20 年間に及ぶキャンペーンの幕が閉じました。[83,84]

PREEMPT_RT の取り組みは 2004 年、Gleixner、Ingo Molnar、Steven Rostedt、そして故 Doug Niehaus によって開始され、Linux において、応答時間を一定の範囲内に収め、予測可能にすることを目指しました。故 Daniel Bristot de Oliveira も特にデッドライン スケジューリングのコードで大きくコントリビューションを行いました。最後の仕上げとして、Linutronix のエンジニア Sebastian Andrzej Siewior と John Ogness が仕事の多くを担いました。Ogness はコンソール出力がリアルタイム処理を停滞させないようにカーネル printk ロギングを再構築し、最後の主要なブロッカーを解決しました。[85]



出典：「リアルタイム プリエンプションのプルリクエスト」*LWN.net.* 入手先 : **https://lwn.net/Articles/990985/**

コードが難しかったわけではありませんでした。20 年間に及ぶ継続的なインフラ構築作業への資金調達こそが困難でした。ドイツの OSADL が初期の業界サポートを組織しました。2015 年、Linux Foundation は Real-Time Linux コラボレーティブ プロジェクトを立ち上げ、[86] Gleixner を Linux Foundation Fellow に指名し、アップストリーミングを完了するためにメンバー資金を集約しました。プロジェクトの全期間におけるメンバーには、Altera、ARM、BMW、CIP、ELISA、Ericsson、Intel、IBM、Google、National Instruments、OSADL、Red Hat、Siemens、Texas Instruments が含まれます。

この取り組みにより、クラウドサーバーから、マイクロ秒の精度を要するデバイスに至るまで、あらゆるサイズのシステムで動作する単一の Linux コードベースがもたらされました。Red Hat Enterprise Linux for Real Time、SUSE Linux Real Time、Ubuntu のリアルタイム カーネルなどの商用製品は、現在、産業用ロボティクス、Software-Defined Vehicle(ソフトウェア定義車両)、5G 無線ネットワーク、医療機器、航空宇宙・防衛システム、プロフェッショナル オーディオ機器に提供されています。ELISA プロジェクトが安全認証を進める中、リアルタイム Linux はエッジ環境におけるロボット技術とフィジカル AI の次の波に向けてポジションを得ています。

リアルタイム Linux はメンターシップの物語でもあります。Anna-Maria Behnsen はキャリアの初めに、Linutronix に就職して同社のリアルタイム チームに配属され、彼女の父親 Thomas Gleixner をメンターとして Linux カーネルを学びました。彼女はカーネルのタイマー領域の専門家の一人となり、タイマーサブシステムが CPU をまたいでタイマーの期限切れを処理する仕組みを再設計しました。2024 年にはカーネルの高精度タイマーとタイマーホイールの共同メンテナーに指名されました。[87] Linux の正確な時間管理を支えるコードは、Linux を時流に乗せて、今や第二世代へと引き継がれています。

エッジ

エッジコンピューティング システムにおいて、データは多様な物理システムから収集され、それらすべてをクラウドに送信するのを回避するために、できるだけローカルで処理されます。このハードウェアの多様性とローカル システムにおけるリアルタイム処理には、Linux オペレーティングシステムが備えるモジュラリティ、決定論的動作、およびコンテナ化の機能を必要とします。[88] Linux カーネルは、ハードウェアの可搬性、マイクロ秒単位での完全な実行時間制御、制約のあるハードウェア上のクラウドネイティブ、およびオープンライセンスによるシステム拡張を可能とします。

> **今日、世界中のエッジデバイスの 75% 以上が Linux カーネル上で動作しています。[89]**

Linux はエッジコンピューティングの推進において重要な役割を果たしてきました。「エッジ」が存在する前、組み込みシステム開発者たちは、プロプライエタリーの専用リアルタイム オペレーティングシステム（RTOS）のロイヤリティを回避するために Linux へ移行しました。Linux を使用すれば、軽量でカスタマイズされたカーネルを構築するために不必要なドライバーを取り除くこともできました。[90] 組み込みシステムでの Linux の普及は、**Yocto Project** や **LF Edge** など、オープンソース プロジェクトのより広範なエコシステムを生み出しました。カーネルがコンテナ化に向けて進化すると、開発者は、ハードウェアがノート PC であれ、クラウドサーバーであれ、産業用ゲートウェイであれ、アプリケーション ソフトウェアを同一実行環境で動作させるツールを得ました。

エッジコンピューティング エコシステムが Linux を中心に進化するにつれ、カーネルも同様にエッジコンピューティングのニーズを満たすために進化しました。たとえば、PREEMPT_RT のアップストリーム化により、業界は Linux カーネルが具備する信頼性の高い品質プロセスを活用できるようになりました。これにより、決定論的な動作が求められる重要なアプリケーションにおいて、分析作業を簡素化できるようになりました。その結果、医療機器など遅延に敏感なエッジ環境での Linux 採用が増加し、2025 年にはヘルスケア向けエッジで最も急速に成長する売上シェアを記録しました。Ubuntu Core や Alpine Linux など、最小限の低リソースシステム上で動作するよう最適化されたエッジ向け OS も登場しました。[91] 携帯電話基地局や地域データセンターなどのより大きなエッジインフラにおいては、Linux の eBPF などのツールはローカル エッジノードが高スループット トラフィックを高速に検査・ルーティングし、処理オーバーヘッドをバイパスすることを可能にします。[92]

エッジコンピューティングは IoT、エンタープライズ、テレコム、およびクラウドエッジにわたって進化してきました。Linux カーネルを置き換えるのではなく、多数のプロジェクト（EdgeX、EVE、Open Horizon、Akraino）はその機能を拡張し、産業用 IoT、小売り、製造、スマートシティ、電気通信、および AI にわたってイノベーションを加速するオープンでベンダーニュートラルなエコシステムを創造しています。エッジインフラが AI 推論とリアルタイム アナリティクスのサポートを強化するにつれ、Linux カーネルは異種ハードウェアとクラウドネイティブ ソフトウェアを接続する信頼される実行環境として機能し続け、オープン エッジコンピューティング エコシステムの礎石となっています。

Linux カーネル開発者シリーズ：Mark Brown



Mark Brown の Linux への参入は 30 年前、Debian 開発者として始まりました。彼の最も初期のカーネルへのコントリビューションはネットワークスタックでした。彼はパフォーマンス問題の修正パッチをアップストリームに提供し、またネットワークコントローラ上の深刻なハードウェアバグの回避策を開発しました。当初は、勤務先で手がけていた組み込み製品に関連する業務の一環として、パートタイムでカーネルのアップストリーム開発に携わっていました。その後、Wolfson Microelectronics（チップメーカー）のフルタイム カーネル開発に移り、同社で Wolfson デバイスのドライバーコードを書き、それを保守していました。彼は後に Linaro、そして Arm へと移り、常にアップストリームに焦点を当てた役割を担い続けています。

Mark は複数のサブシステムにわたるメンテナーであり、ALSA、テスト実行、および regmap にコントリビューションしています。彼はまた、Stephen Rothwell の退任に伴い、メインラインマージ以前にパッチをテストするための主要なインテグレーション ツリーである Linux Next を管理しています。この役割において、彼は LLM（大規模言語モデル）を用いて作成された AI 支援パッチの増加と、これらがメンテナーの負荷をいかに大幅に増大させているかを証言しています。彼は、AI 支援メンテナンスに対するコミュニティの取り組みやすさにもかかわらず、これらのツールへの信頼が疑問視されていることを説明します。「Coccinelle のようなツールは広範なコード変換を本当に行いやすくします…それらはすべて決定論的で規則的です…しかし、LLM では、予測可能性が遥かに低くなります…『はい、私はこのツールを信頼する、このツールはたぶん正しいことをするだろう』という同じ感覚を持つことはできません。」彼は、多くのコードへのコントリビューション以外に、Linux Plumbers Conference の組織委員会委員長を務めています。

なぜ関わり続けることを選ぶのかについて、彼は「テクニカルな挑戦と、人々にとって可能な限り最高の仕事をする何物かを作ろうとすることの組み合わせです。私たちが一緒に働いている人々の存在が鍵となる部分です」と語りました。

Mark のインタビュー全編は、**My Life as a Linux Kernel Developer and Maintainer with Mark Brown** [93] を、是非ご視聴下さい。

人工知能

現代の人工知能は Linux カーネル上で誕生しました。Linux は、新しい AI ワークロードを支えるために必要な高スループット・低遅延の基盤を提供します。AI ツールと自動エージェントは同時に、カーネル自体の開発作業、セキュリティ メンテナンス、およびコードレビュー ワークフローを変革しています。

グローバルなワークロードインフラの基盤として、Linux カーネルは、機械学習ワークロードに存在する極端な揮発性と並列性の高さをサポートするために、急速にテレメトリ駆動型の近代化を行っています。たとえば、Linux 6.12 で導入された sched-ext フレームワークは、AI を含む多くのタイプのワークロード向けに特化したプログラマブル スケジューラを可能にします。このフレームワークにより、ワークロードは Linux の旧来のスケジューラの硬直性をバイパスし、代わりに安全な BPF プログラムとしてカスタマイズされたポリシーを実行し、それにより CPU レベルでのボトルネックを削減し、時には敏速な実験を許容することもできます。[94,95] AI エージェントはまた、ワークフロー全体で突発的なリソース使用量の変動、－例えば、コードファイルを読み込む際は最小限のメモリを使用するが、呼び出しを実行する際は膨大な使用量となる－を管理するために、カーネルにより大きな柔軟性を必要としています[96]。AgentCgroup などの新しいフレームワークは、変動に応じて、リソースの割り当てや管理を、きめ細かく設定・管理する仕組み、グラニュラー リソース バジェット機能を備えており、カーネル内での柔軟なメモリ割り当てを可能にします[96,97]。これらの新しいフレームワークは、リアルタイム実行テレメトリに基づいて、カーネルが動的にリソース割り当てを伸縮するのを助けます。

AI ワークロードのコンピューティング処理能力ニーズはまた、半導体業界に一社単独の孤立したハードウェア製造から、Linux カーネルおよびオープンソース エコシステムとのアーキテクチャ上の共同設計へと軸足を移すことを強制しました。AI エージェントのワークフロー、例えば複数のツールを呼び出し、メモリ内に情報を保持し、サブエージェントを作成し、意思決定を行うというケースを考えると、単一のプロセッサでそのワークフローを処理するのは非効率であることが分かります。その結果として、彼らは共同設計を通じて調整された特別なハードウェアを必要としています。そこにおいては、半導体製造業者がオペレーティングシステム カーネルと同期しながら自社のチップアーキテクチャを最適化します。[98,99] 関連して、AI ワークロードの要件が汎用プロセッサ

設計を分岐・多様化させるにつれ、半導体製造業者は、低コストで、特定のカスタマイズされたワークロードにチューニングされたプロセッサの開発を可能にする RISC-V などのオープン標準命令セットアーキテクチャ（ISA）を製品に組み込んでいます。[100]

半導体エコシステムは生成 AI の計算需要を満たすために同期しており、製造業者はプロプライエタリーなハードウェア開発から、オープンソース Linux カーネルとの統合、およびオープン標準アーキテクチャへとシフトしています。自社のシリコンがグローバルなクラウドネットワーク全体でスケールできることを保証するためです。

AI の技術的要件とは別に、Linux カーネルは AI ツールの影響を考慮して変化しなければなりませんでした。他のあらゆるソフトウェア エンジニアリング活動同様に、Linux カーネルはますます多くの AI 支援コードパッチを受け取っており、Linux コミュニティは 2026 年 4 月に AI ツールの使用に関するポリシーをリリースし、AI ツールを帰属表示と人間のサインオフを必要とする支援ツールと位置付けることで AI ツールの使用を規約化しました。[101] ポリシーは、AI 支援パッチであることがより良く判別できるようにするために作成されましたが、しばしば重複し、冗長な修正レポートが開発パイプラインを詰まらせるため、大量のパッチの管理は議論継続中の懸念事項となっています。[102] 一方で、メンテナーの間では、コードレビューのための AI ツールがカーネル コミュニティで支持を得ています。その一例が Sashiko プロジェクトであり、パッチを受け取り、レビューチームを模倣したレビュープロトコルに基づいてそれらを評価するエージェント ツールです。[103] 2026 年 7 月、Torvalds は Sashiko に言及し、メーリングリストでカーネルによる AI ツールの使用を承認しました。「Linux はアンチ AI のプロジェクトではありません…AI はツールであり、私たちが使用する他のツールと同じです。そして明らかに有用なツールです。[104]」AI ツールが進化するにつれ、それに対するカーネル コミュニティの応答は疑いなく他のプロジェクトのポリシーや行動に影響を与え続け、プログラミングやバグ修正における AI 採用の波及効果を生み出します。

他のあらゆるオープン ソースプロジェクトと同様、AI の普及によって脅威の状況が拡大したことを受け、カーネルのセキュリティを再評価する必要性が生じています。悪意ある攻撃者は、AI ツールを駆使して公開されているコードベースを探索し、複数のバグを組み合わせることで、複雑なエクスプロイト（攻撃手法）を作り出しています。これに対抗するため、セキュリティ メンテナー側も同様のツールを防御目的で活用し、未修正のまま放置されているバグのパターンを特定した上で、パッチや修正プログラムを生成しています[105]。現在、IBM と Red Hat による Project Lightwell や、Linux Foundation の Akrities といった取り組みに見られるように、企業ステークホルダーが結束して、この新たな脅威の状況に対処しようとしています。[106,107]

Linux カーネル開発者シリーズ：Sasha Levin



Sasha Levin はオープンソース ソフトウェア開発への自身の道を、Windows と Linux の両方で動作するネットワークドライバーを書く仕事を課されたときに始まったと説明します。Windows での作業はドキュメントを精査し、回避策を探すことを必要としたのに対し、Linux ではソースコードを直接見て問題を特定し、修正を送ることが可能でした。この体験により彼はフルタイムの Linux カーネル開発へと移行し、約 15 年前に KVM ツール（軽量ハイパーバイザ）の構築を開始しました。時間とともに、彼はコアカーネル作業、ライブパッチ適用、および Linux 安定リリースの保守へと引き寄せられました。さらに、Sasha は生成 AI が普及する遥か以前から自動化ツールをカーネルに適用しており、安定リリースへの修正を特定し、メインライン化するためにこれらのツールを使用していました。「AI はコードを理解し、なぜ特定のやり方でものが行われているのかを理解するための素晴らしいツールになり得ると考えます。」と彼は語りました。

NVIDIA での現在の役割において、彼は Linux カーネルの安全認証の取り組みを率い、それに必要な機能をアップストリーム化しています。Sasha にとって、会社内で正しい関係と役割を見つけることは、彼のメンテナーとしての負荷を管理するために極めて重要でした。「アップストリームでの私の作業が、オープンソースに対して行っている寄贈行為や慈善事業と見なされないことを本当に望んでいました。会社にとっても、自分にとっても、そしてカーネル世界にとっても有益であってほしいと望んでいました…つまり、自分のキャリアを、自分が心から愛する活動を中心に構築するということなのです。」

Sasha のインタビュー全編は、**My Life as a Linux Kernel Developer and Maintainer with Sasha Levin** [108] を、是非ご視聴下さい。

持続可能性とカーネルの未来

この驚くべき影響力とともに、明確な質問が Linux コミュニティに常に存在しています：私たちはどのようにしてカーネルの持続可能性と未来を保証するのか？35 年を経た今、非常に健全なオープンソース プロジェクトとしての証拠が存在します：各リリース サイクル毎に数百人の新規コントリビューターが現れ、開発者の半数以上が四半期を超えてコントリビューションを続けています（**図 1, 3** を参照）。コードベースは、拡大と縮小を繰り返しており、機能が追加されるのと同じくらい削除されていることを示しています。しかし、AI の台頭によって新たな脅威が浮上していること、そして極めて重要なメンテナーの育成・確保という課題を考慮すると、コミュニティは今後も、戦略的計画の最優先事項として持続可能性を据え続ける必要があります。

セキュリティ

Linux カーネルは至る所に存在しているため、Linux のセキュリティは世界中の IT スタック全体に非常に重大な影響を持っていします。2015 年、プロアクティブに構造的防御を Linux カーネルに統合するために Linux Kernel Self-Protection Project (KSPP) が立ち上げられました。その目標は、未知の脆弱性が発生したときに安全に停止（fail safely）するよう Linux カーネルを設計することです。たとえば、KSPP は極めて危険なことで知られる strcpy 関数（終端記号に達するまでデータを無差別にコピーし、壊滅的なメモリ破損を招く恐れがある）を体系的に排除し、転送データ量を厳密に制限する strscpy のような代替関数へと置き換える取り組みを主導しました。同様に、プロジェクトは論理的 case 文での偶発的な「ドロップスルー（フォールスルー）」を拒否するようビルドシステムを改変しました。過去には、開発者が break 命令を忘れることで、プロセッサが誤って間違ったコードブロックを実行する原因となり得ました。今日、開発者は意図的なドロップスルーを明示しなければならず、危険な偶発的欠落を防ぎます。究極的に、これらのセルフプロテクション メカニズムは、ソフトウェアが実行される前にエラー カテゴリー全体が排除されることを保証します。

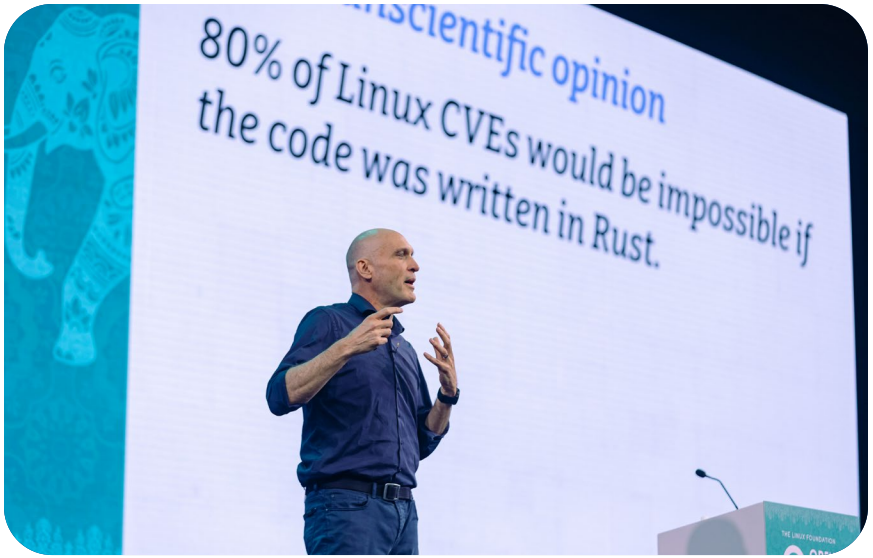
現代コンピューティングのメモリ安全性危機（多くのシステムにおけるほとんどの脆弱性がメモリ安全性エラーであるという認識）は、セキュリティを重視するコミュニティ メンバーに、カーネルの一部を Rust で記述するという重大なタスクを試行させました。Rust が選ばれたのは、C 言語にない厳格な型チェックと借用チェック（borrow check）を通じてコンパイル時にメモリ安全性を保証し、ランタイム ガベージコレクターのパフォーマンス上のペナルティを回避するからです。Rust の使用は、それらのカーネル部分の類似した脆弱性の可能性を排除しながら、C 言語のパフォーマンスを維持します。[109] 2020 年、Rust for Linux プロジェクトは、この言語が既存の C コードと共存できるかどうかをテストし始めました。最初の Rust インフラは 2022 年にマージされましたが、C や C++ 以外の高級言語でプロダクション カーネルの重要な部分をビルドする試みが何十年も行われていなかったことを考慮したために、実験的な試みとみなされていました。[110] そこから、最初のプロダクション Rust ドライバーがマージされ、メンテナー コミュニティにおける多少の軋轢にもかかわらず採用が増加しました。[111,112] 2025 年後半、コミュニティは Rust がもはや実験的ではないことを発表しました。[113] Greg Kroah-Hartman が語ったように、Rust を採用するという決定は、開発者体験を向上させることよりも、むしろレビュー担当者の作業負荷を軽減する必要性に突き動かされたものでした。[109]

Linux カーネルは、厳格な基準を満たすことにより、オープンソース ソフトウェアに対する最高レベルのセキュリティ セルフ認証である OpenSSF Best Practices Gold Badge を獲得しました。[114]
このマイルストーンを達成しても、カーネルは新たな脅威や攻撃から身を守るために進化し続けなければなりません。オープンソースコミュニティ自体のセキュリティが厳しく問われる事態となったのが、XZ Utils をめぐる攻撃でした。この事例では、国家の関与が疑われる攻撃者が、XZ Utils の燃え尽きたソロ メンテナー（一人きりでメンテナーの重責を担っている）をターゲットにして彼の信頼を獲得し、バックドアを注入しましたが、コミュニティのエンジニアによって偶然発見されました。[115]これはソーシャル エンジニアリング攻撃を防ぐ必要性を浮き彫りにし、またソロ メンテナーによって運営されるプロジェクトのガバナンスの転換と、より厳格なアクセス制御を促しました。 [116] コードレビュー

プロセスも、前述のセクションで議論したように AI 生成パッチが Linux カーネル メーリングリストを含むオープンソース ソフトウェアのメンテナーに殺到する中で、新しいプレッシャーに晒されています。

これは、悪意ある攻撃を避けるために脆弱性の非公開での共有を意図した Linux カーネルの非公開セキュリティ リスト (セキュリティ チーム、メンテナー、主要ディストロで構成) にとって特に問題となっています。この種の報告の大量流入は、多数の冗長または重要でないセキュリティパッチによってセキュリティ リストをほぼ管理不能にしています。この殺到に対処するため、Linus は AI ボットが報告しているものは何であれ、おそらくすでに公開されているため、セキュリティ リストに報告すべきではないと発表しました。2026 年春、コミュニティはこれに対処するためにカーネル ドキュメントにいくつかの更新を行い、何がセキュリティバグに該当するか、脆弱性の定義と脆弱性に当たらないものの定義、通常の開発プロセスで何を扱うべきか、および非公開セキュリティ リストで AI ツールを適切に使用する方法を明確にしました。[117]

Linux カーネル開発者は他の開発者の体験やツールを学習・活用し、その見返りとして、他の人々のために規模を拡大して脆弱性を発見し、それに対抗する方法を実証してきました。Linux Foundation の Open Source Supply Chain Security のディレクター David A. Wheeler が説明するように、「35 年間の Linux カーネル開発は、世界規模のセキュリティを実現するにはオープンなコラボレーションと妥協のない誠実さが必要であることを証明しました。Linux メンテナーは狭い脆弱性の定義の陰に隠れたり、希少事例を絨毯の下に押し込んだりしません。Linux カーネル コミュニティはコードを注意深く検査し、さらに再検査して、すべてのユーザーが安全であり続けられるよう、保守的だとの批判を恐れることなく、多くのソフトウェア欠陥を潜在的セキュリティリスクとして扱っています。」



Open Source Summit India で講演する *Greg Kroah-Hartman*、2026 年 6 月

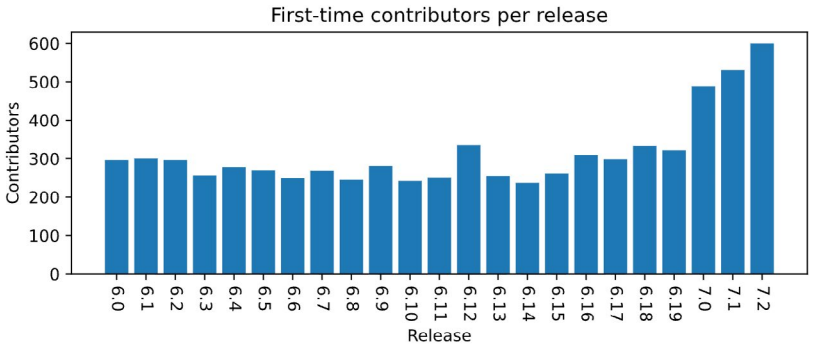
次世代のためのメンターシップ

Linux カーネル コミュニティは、自発的かつ主体的な参加と関心のモデルを通じて機能します。コントリビューターの歩みは、カーネル メンテナーであり Linux Fellow でもある Shuah Khan が「道（road）」と表現するものに従います。そこでは開発者の関与の程度は個人の主体性、関心、および利用可能な時間に依存します。Khan は「オープンソースでは、新しい開発者をリクルートし、それらのリクルートされた人がメンテナーになっていくというわけではありません。自発的にやって来て、そこに留まり、また留まるか去るかを選択する人々のコミュニティなのです。誰かが開発者としてのあなたをパイプラインに押し出すわけではありません。あなた自身がその道を旅する責任者なのです」[118]と語りました。開発者は、長年に渡り頻繁にコントリビューションするようになるかもしれませんし、あるいは移動前に特定の問題を解決するために少数のパッチを提出する、より短期で過渡的な存在かもしれません。継続的なエンゲージメントはコミュニティを構築するために理想的ですが、どのような形のエンゲージメントであっても、オープンソースの手法、技術的標準、および協調的活動をより広いソフトウェア業界へ拡散するのを助けます。

カーネル開発者の将来の世代を構築することは、プロジェクトの長寿を保証するために極めて重要です。カーネルへのコントリビューションは 1991 年から飛躍的に増加し、2014 年にはリリース毎に、平均 2,000 人のコントリビューターによる 12,000 ～ 15,000 コミットで安定し始めました。より最近では、健全な開発者コミュニティとして勇気づけられる兆候が存在します：2,362 人の開発者が Linux 7.0 にコントリビュートし、そのうち 489 人が初参加でした。[119] この増加した新規参入者の数は Linux 7.1 および 7.2 マージでも継続し、AI ツールの使用によって推進されている可能性があります（図 3 を参照）。[3,120] 今日、コントリビューションは米国に大幅に偏っており（25%）、ドイツと中国が 2 位と 3 位となっています。[20] リーダーシップの継承はカーネルの継続性の鍵となる重要な側面であり、2026 年 1 月、コミュニティは Linus の後継者を選択するためのプロセスのドラフトを作成しました。[121]

LF Mentorship Program は、カーネル コミュニティの持続可能性と多様性を支える極めて重要な要素です。オープンソース開発への参加が地理的、言語的、および財務的な制約に沿って不均等に分散しているため、真に多様な参加を得るには参入を阻む構造的な障壁に対処する必要があります。これが、オープンソース開発者コミュニティの多様性を増加させるために Linux Foundation が 2019 年に LF Mentorship Program を設立した主な理由でした。[122] プログラムにはフォーマルなメンターシップ プログラムだけでなく、集中的な学習をサポートするためのより柔軟な学習機会や手当支給が含まれています。[122,123] **LF Live Mentorship and Maintainer series** は、キャリアパスについてコミュニティの主要メンバーにインタビューし、特定の技術開発領域への取り組みについて語ってもらっています。これは、専門家や学習者の時間を過度に奪うことなく、知識を共有・伝達する手段となっています。[124,125] ウェビナーやメンターシップは、本来の開発スキルを超えて、信用を構築する方法、社会規範を守る方法、およびコミュニティの信頼を確立する方法を含む、コーディングの社会的側面を教えます。2019 年 1 月から 2026 年中旬までに、プログラムには 111 カ国以上から（圧倒的多数がインドから）6,447 人以上の応募者があり、73% が男性、14% が女性、0.8% がノンバイナリーと自認しています。[126]

図 3: Linux 6.0-72 の各リリースにおける初参加のコントリビューター



出典：「*Development statistics for the 7.2 kernel*」、*LWN.net*。入手先：**https://lwn.net/Articles/1088776/** [3]

生成 AI の導入は、カーネル開発における最新の転換点となっています。こうしたツールは、定型コード（ボイラープレート）の生成、技術文書の草案作成、さらには複雑なサブシステムを調査する手助けなど、開発の初期段階において大きな恩恵をもたらしています。しかし、コード作成を自動化ツールに依存することには、長期的な成功と習熟のために不可欠な根本的なソフトウェアエンジニアリング経験と深い意味論的な理解をバイパスするリスクがあります。AI ツールの使用はメンターシップの人間的要素も減少させます。Khan は次のように説明しています。「大規模言語モデル（LLM）は技術を教えてくれるかもしれないが、オープンソースの本質は単なる技術に留まりません。特に『人間』に関わる側面は、うまく通り抜ける道を見つけるのが難しく、習熟するのにも時間がかかるものです。そうした側面は、メンターとの対話を通じてこそ学べるものなのです」。[118] オープンソース コミュニティの将来的な強靭性は、開発者がオープンソースにおける人間的な側面をどのように見つけるかを学ぶために、必要な対話の場や経路をいかに維持できるかにかかっています。

> **コード作成を自動化ツールに依存することには、長期的な成功と習熟のために不可欠な根本的なソフトウェア エンジニアリング経験と深い意味論的な理解をバイパスするリスクがあります。**

この人間的な側面は、Khan がプロジェクトに意欲的に取り組み続けている重要な理由の一つです。「世界中の人々と一緒に働くことができるという意味でこのような満足を与えてくれるものは、世界の他のプロジェクトにありません。」[71]

Linux カーネル開発者シリーズ：Greg Kroah-Hartman



Greg Kroah-Hartman は、カーネル内の USB 仕様の問題を修正しようと試みている時に、組み込みシステムの世界で、Linux メンテナーとしての自身の旅を始めました。自身の最初のコントリビューションを思い出して、彼は余暇としてプロジェクトに取り組み始めたことを語りました。「私がメーリングリストに投稿した最初メールは、『パッチをどう書くのか?』という質問でした。良いドキュメントがなかったので、パッチの書き方が分からなかったのです。」彼は、いかに経験豊富なコミュニティ メンバーと協力し、彼らの直接のフィードバックを吸収したことが自分をより良いエンジニアにしたかを説明しました。長年にわたり USB メンテナーとして活動してきた Greg は、カーネル メンテナーシップを伝統的な仕事とは異なるものとして説明しました。メンテナーが、雇用主や仕事上の役割を変えた時にも、自らの知識とサブシステムに対する責任を持ち運ぶためです。彼はこのユニークさを「メンテナーを雇うことはできません。彼らがメンテナーへと育たなければならないのです」と述べて要約しました。

異なるサブシステムにわたる多様な負荷を管理するため、Greg は自身の役割の優先順位付けの重要性を説明しました。週の早い段階で安定リリースをレビューし、週半ばまでにパッチを待ちキューに入れて Linus に送り、Rust での作業を進め、USB コントリビューションを管理する——そしてそのすべてにおいて、少し離れた立ち位置を取り、委任する方法を学ぶことです。カーネルを保守するために数十年にわたって構築された知識は、重要な技術的パターンを認識するために必要な専門的な知見を提供しており、これはコミュニティの AI パターンマッチング作業にとっても有用です。なぜこの仕事をやり続けるのか、そしてそこから得る満足感について振り返り、Greg は「カーネルを一度だけ作ることは実に簡単です。誰にとっても安定し、正しく動作するものを、時間を超えて、いつも絶えず、繰り返し行うことは本当に難しいことです。それは驚くべきエンジニアリングの偉業なのです。」と総括しました。

Greg のインタビュー全編は、**My Life as a Linux Kernel Developer and Maintainer with Greg KH** [127] を、是非ご視聴下さい。

カーネルが人々に与えたインパクト

> **カーネルを一度だけ作ることは実に簡単です。誰にとっても安定し、正しく動作するものを、時間を超えて、いつも絶えず、繰り返し行うことは本当に難しいことです。それは驚くべきエンジニアリングの偉業なのです。**
>
> **– Greg Kroah-Hartman, Linux Fellow**

グローバルなコンピューティング環境のために高いパフォーマンスとモジュラリティを備え、無償かつ自由に利用できる基盤を提供することにより、Linux カーネルは経済的平等、グローバルなコラボレーション、ソフトウェア ガバナンス、およびデジタル主権のあり方を大きく変えてきました。Linux 自体の産業的および技術的影響を超えて、カーネルとオープンソースのエコシステムは、世界中の人々の日々の生活のあり方を新たに形作り、その生活を可能にしてきました。オープンソース上に構築され可能にされた高トランザクション金融システム（インドの UPI モバイル決済インフラや PayPal など）は、小規模ビジネスのオーナー、独立ベンダー、および農村コミュニティが携帯電話上の QR コードとアプリだけでデジタル取引を実行することを可能にします。これはインド経済の重要な部分を支えており、2025 年 6 月には、UPI を通じて 180 億件を超える取引が行われ、決済総額は 2.4 兆ルピー（約 250 億米ドル）に達しました。[128] 農業において、オープンソース コントローラは小規模農家が土壌水分を監視しローカルな天候パターンを予測することを可能にしています。エネルギー グリッドにおいて、小さな Linux 駆動コントローラは地域の発電量を管理するために風力タービンなどのグリーンエネルギー インフラを操作します。オープンソースは世界的コミュニケーション プラットフォームの背後にあるデータセンターやサーバーを動かし、遠距離の家族や友人が接続することを可能にし、世界的危機の中でも全世界の科学コラボレーションを可能にします。

Shuah Khan との会話の中で、彼女は次のように語っています。「オープンソースのない世界がどのようなものになっていたかを考えます…そしてオープンソースなしで私たちは何をしていたでしょうか？」開発活動が分散的かつ自由に行われるという特質のために、新たな視点やアイディアがもたらされ、クローズドソースの環境では実現し得なかった進歩の形や方向性が生み出されました。彼女は続けます。「Linux――Linux とオープンソース――が世界を動かしています。これは決して軽々しく言っているわけではありません。そして、そこにこそ素晴らしさがあるのです。人々は Linux の存在すら知らないかもしれませんが、私たちは世界に対して極めて大きな影響を与えているのです」。このエコシステムは大きな勢いを持っていますが、オープンソースのオペレーティングシステムやその他のプロジェクトが、必然的に生まれたわけではありません。それらが成立するには、適切なタイミングで適切な人材が揃うことが不可欠であり、それは今後も変わりません。このことは今日においても重要な意味を持っています。オープンソースの AI モデルをめぐる現在の緊張関係は、2000 年代初頭の企業向けオペレーティングシステムをめぐる「OS 戦争」を彷彿とさせるからです。この比較は、歴史から学べる明確な教訓を示しています。すなわち、オープンソースという選択肢は、常に企業間競争の前の段階における安全で、カスタマイズ可能で、かつ費用対効果の高い基盤となり、世界規模のエコシステムがその上に構築されていくのです。

おわりに

趣味のプロジェクトとしての控えめな芽生えから 35 年を経て、Linux カーネルは疑いなく現代のグローバルインフラの根幹をなすエンジンとなりました。Linux のモジュラー アーキテクチャと分散開発モデルはソフトウェア エンジニアリングを革新しました。その産業的影響は事実上すべてのドメインに及び、SDV(ソフトウェア定義車両)、ハリウッドのレンダリング パイプライン、低遅延の金融取引所、安全重視の医療機器、および宇宙でのミッションクリティカルな運用を可能にし、それを支えています。技術的に、Linux はエンタープライズメインフレームやサーバーから進化し、クラウドネイティブコンテナ、高スループットネットワーク、およびエッジコンピューティングノードのための不可欠な実行環境となりました。

未来展望として、Linux は現代の AI などの新興テクノロジーに影響を与え、加速させるユニークな立ち位置にあります。AI が前例のない計算スケールを要求するにつれ、カーネルは複雑な機械学習ワークロードと専用シリコンアーキテクチャをオーケストレートするためにその中核的な機構の進化、近代化を続けています。同時に、Sashiko などの AI ツールやレビュー プラットフォームはカーネル メンテナンスとバグ修正のあり方を作り変えています。Rust 統合や厳格なコミュニティ ガバナンス ポリシーなどのプロアクティブなセキュリティ イニシアチブに支えられて、Linux コミュニティは急速なイノベーションがシステム的な安定性を決して損なわないことを保証しています。

Linux カーネルが 20 周年を迎えたとき、IBM Fellow の Irving Wladawsky-Berger はこのマイルストーンを振り返り、Linux をこれほど重要なものにしている理由を見失わないよう Linux 愛好家たちに呼びかけました。彼が語るように、「Linux について非常に貴重なものはコミュニティです。それこそが私たちが絶対に失ってはならない一つのものなのです。私たちのオープンエコシステムを維持し続けることは非常に重要であり、そうすることで私たちの社会が直面している最も重要な問題のいくつに対処する役割を果たし続けることができるのです。」[70] 彼の言葉は今日もなお真実として響いています。



Linux Storage Filesystem MM & BPF Summit の参加者、2026 年 5 月

免責条項

LF Research は、二次資料の集約、データ合成、レポート原案作成支援、原稿編集、翻訳など、リサーチ関連業務において生成 AI ツールをさまざまな度合いで活用しています。AI が作成したすべてのコンテンツは、人間が確認し、必要に応じて修正を行っています。

この日本語文書は、**35 Years of Linux** の参考訳として The Linux Foundation Japan が提供するものです。

謝辞

著者らは、レポートの執筆にあたり、データ、査読、および有益な指導を提供してくれた本レポートへの多数の貢献者に感謝を表したいと思います。特に以下の個人の方々のサポートに感謝いたします：

- Chris Aniszczyk
- Mark Brown
- Gabriele Columbro
- Jonathan Corbet
- Mike Dolan
- Tosha Ellison
- Dirk Hohndel
- Arpit Joshipura
- Shuah Khan
- Greg Kroah-Hartman
- Julia Lawall
- Sasha Levin
- Paul E. McKenney
- Walt Miner
- Ramon Roche
- Steven Rostedt
- Konstantin Ryabitsev
- Kate Stewart
- Linus Torvalds
- David A. Wheeler
- Irving Wladawsky-Berger

著者紹介

ANNA HERMANSEN は、Linux Foundation のシニアリサーチャー兼エコシステムマネージャーであり、そこにおいて研究プロジェクトを率い、Linux Foundation の研究のエンド ツー エンドの管理をサポートしています。彼女はヘルスデータ インフラやオープンソース AI において定性的および体系的レビュー研究を実施し、カンファレンスやワーキング グループで研究発表を行ってきました。彼女の関心は AI、プレシジョン メディシン、およびヘルスデータ共有の交差にあります。彼女はクライアント サービス、プログラム デリバリー、プロジェクト マネジメント、および学術・企業・Web ユーザー層向けの執筆体験を持つジェネラリストです。Linux Foundation 以前は、2 つの異なる研究プログラム（Blockchain Research Institute および BC Cancer's Research Institute）に携わってきました。University of British Columbia から公衆衛生学の修士号と国際関係論の学士号を取得しています。

HILARY CARTER は、2021 年に Linux Foundation に入社し、オープンソースのトレンド、機会、および挑戦に対する実証的インサイトを提供するために設立された LF Research を立ち上げ、それを主導しています。Linux Foundation 以前は、ブロックチェーン技術に焦点を当てたトロント拠点のグローバルな協同シンクタンク Blockchain Research Institute のマネージング ディレクターを務めていました。彼女はオープンソースイニシアチブと業界全般でのその採用に焦点を当てた 200 近くの研究プロジェクトに貢献してきました。London School of Economics から経営学の修士号を取得しており、カナダとアイルランドの二重国籍を有しています。

DANIEL M. GERMAN は、University of Victoria のコンピュータサイエンス教授です。彼の研究はソフトウェア エンジニアリング、特にソフトウェアの進化、オープンソース ソフトウェア、知的財産権がソフトウェア エンジニアリングに与える影響、およびオープンソース ソフトウェアが開発・維持・統治される社会的・技術的プロセスに焦点を当てています。より最近では、生成 AI や AI コーディング エージェントがオープンソース ソフトウェア開発に与える影響を調査しており、コントリビューション、コードレビュー、およびメンテナーシップの経済性を彼らがどのように変えるか、ならびにオープンソース プロジェクトにおける説明責任、正確な履歴、およびガバナンスの問題に特別な注意を払っています。彼はアクティブなオープンソース コントリビューターでもあり、プログラマブルな開発環境や開発者がソフトウェアシステムと相互作用する方法を強化するツールに特別な関心を持ってソフトウェアツールを開発・維持しています。

参照資料

1. Sandberg KD. The Value of Open Data on Global Entities [Internet]. The Linux Foundation; 2026 [cited 2026 Aug 27]. Available from: https://www.linuxfoundation.org/research/open-data-on-global-entities
2. Digital Applied [Internet]. 2026 [cited 2026 Aug 27]. Mobile OS Market Share 2026: iOS vs Android Statistics. Available from: https://www.digitalapplied.com/blog/mobile-os-market-share-2026-ios-vs-android
3. Corbet J. Development statistics for the 7.2 kernel. LWN.net [Internet]. 2026 Aug 17 [cited 2026 Aug 27]. Available from: https://lwn.net/Articles/1088776/
4. [MAINTAINERS SUMMIT] Coping with the new-developer flood [Internet]. [cited 2026 Aug 27]. Available from: https://lore.kernel.org/ksummit/CANiq72m8se6AYJu_qDc1T=nCEank25nAu2ooZSox_1VQg7SOtQ@mail.gmail.com/T/#m05b386da142a5ce647555b847ae6e7e9a3d0ba75
5. CNCF [Cloud Native Computing Foundation]. Keynote: There’s Nothing to Fear About the EU’s New Cybersecurity Law - Greg Kroah-Hartman (ASL) [Internet]. 2025 [cited 2026 Aug 27]. Available from: https://www.youtube.com/watch?v=LOT3BccrhpA
6. Martin B. opensource.com [Internet]. 2016 [cited 2026 Aug 27]. What the history of open source teaches us about strategic advantage | Opensource.com. Available from: https://opensource.com/open-organization/16/10/history-open-source-teaches-strategic-advantage
7. TED. The Mind Behind Linux | Linus Torvalds | TED [Internet]. 2016 [cited 2026 Aug 27]. Available from: https://www.youtube.com/watch?v=o8NPllzkFhE
8. Wirzenius L. The early days of Linux. LWN.net [Internet]. 2023 Apr 12 [cited 2026 Aug 27]. Available from: https://lwn.net/Articles/928581/
9. Welcome to LWN.net [LWN.net] [Internet]. [cited 2026 Aug 27]. Available from: https://lwn.net/
10. Çintay E. The Evolution of Linux Kernel Development: A Journey Through Time. Medium [Internet]. 2024 Feb 22 [cited 2026 Aug 27]. Available from: https://medium.com/@Chinacolt/the-evolution-of-linux-kernel-development-a-journey-through-time-01dd2846ada6
11. Linux Kernel Newbies [Internet]. 2024 [cited 2026 Aug 27]. Linux_6.12 - Linux Kernel Newbies. Available from: https://kernelnewbies.org/Linux_6.12
12. Naresh IT [Internet]. [cited 2026 Aug 27]. Linux Architecture Explained Simply Core Guide 2026. Available from:

https://nareshit.com/blogs/linux-architecture-explained-simply-core-guide

13. History of Portability in Linux [Internet]. [cited 2026 Aug 27]. Available from: https://litux.nl/mirror/kerneldevelopment/0672327201/ch19lev1sec1.html

14. The Linux Foundation [Internet]. 2026 [cited 2026 Aug 27]. My Life as a Linux Kernel Developer and Maintainer with Jonathan Corbet. Available from: https://zoom.platform.linuxfoundation.org/webinar/register?project=tlf&id=oXytj4iRU1Zir0hQxx0BRBbjPFpAlAGbuuczJizxsoa0x%2B3FIblVzHiL1Ec8NN2xpBjrTne5L3NZbgO3nuE%3D&logo=thelinuxfoundation-color.svg&zoomID=97950815459

15. cregit. cregit/cregit [C] [Internet]. 2026 [cited 2026 Aug 27]. Available from: https://github.com/cregit/cregit

16. LWN.net [Internet]. 2002 [cited 2026 Aug 27]. linux-2.5.4-pre1 - bitkeeper testing. Available from: https://lwn.net/2002/0207/a/2.5.4-pre1.php3

17. Developer Certificate of Origin [Internet]. [cited 2026 Aug 27]. Available from: https://developercertificate.org/

18. git [Internet]. [cited 2026 Aug 27]. Git - A Short History of Git. Available from: https://git-scm.com/book/ms/v2/Getting-Started-A-Short-History-of-Git

19. Blau T. Git turns 20: A Q&A with Linus Torvalds. The GitHub Blog [Internet]. 2025 Apr 7 [cited 2026 Aug 28]. Available from: https://github.blog/open-source/git/git-turns-20-a-qa-with-linus-torvalds/

20. LFX [Internet]. [cited 2026 Aug 28]. The Linux Kernel Organization Insights. Available from: https://insights.linuxfoundation.org/project/korg?timeRange=past365days&start=2025-08-28&end=2026-08-28

21. Hoffmann M, Nagle F, Zhou Y. The Value of Open Source Software. Harv Bus Sch [Internet]. 2024 Jan [cited 2026 Aug 28]. Available from: https://www.hbs.edu/faculty/Pages/item.aspx?num=65230

22. Software-defined vertical industries: transformation through open source [Internet]. 2020 [cited 2026 Aug 28]. Available from: https://www.linuxfoundation.org/resources/publications/software-defined-vertical-industries-transformation-through-open-source

23. Berger M. CNN [Internet]. 2001 [cited 2026 Aug 28]. IBM brings Linux to Wall Street - August 30, 2001. Available from: https://www.cnn.com/2001/TECH/industry/08/30/ibm.wallstreet.idg/index.html

24. Notes on the Generic Block Layer Rewrite in Linux 2.5 — The Linux Kernel documentation [Internet]. [cited 2026 Aug 28]. Available from: https://www.kernel.org/doc/html/v5.5/block/biodoc.html

25. The Linux Foundation. How Linux Took Over the World of Finance - Christoph H Lameter, Jump Trading LLC [Internet]. 2019 [cited 2026 Aug 28]. Available from: https://www.youtube.com/watch?v=UUOM4KdaHkY

26. LWN.net [Internet]. [cited 2026 Aug 28]. New driver API for Wireless Extensions. Available from: https://

lwn.net/2001/1220/a/wireless-api.php3

27. Grokipedia [Internet]. [cited 2026 Aug 28]. Wireless tools for Linux. Available from: https://grokipedia.com/page/Wireless_tools_for_Linux

28. Chou F. Linux Wireless Networking: a short walk. Linux.com [Internet]. 2015 Mar 15 [cited 2026 Aug 28]. Available from: https://www.linux.com/training-tutorials/linux-wireless-networking-short-walk/

29. Butler S. How-To Geek [Internet]. 2026 [cited 2026 Aug 28]. How one GPL lawsuit accidentally created the homelab router that changed networking forever. Available from: https://www.howtogeek.com/a-legal-accident-turned-this-60-router-into-the-most-legendary-homelab-device-of-the-2000s/

30. OpenWrt [Internet]. [cited 2026 Aug 28]. [OpenWrt Wiki] Table of Hardware. Available from: https://openwrt.org/toh/start

31. Qualcomm Linux system software architecture - System Software Overview User Guide [Internet]. [cited 2026 Aug 28]. Available from: https://docs.qualcomm.com/doc/80-70029-252/topic/qualcomm-linux-sw-overview.html

32. Cisco [Internet]. 2021 [cited 2026 Aug 28]. Cisco NX-OS: The Advanced Network Operating System Designed for the Future. Available from: https://www.cisco.com/c/en/us/td/docs/dcn/whitepapers/nx-os-advanced-network-operating-system.html

33. Roku [Internet]. [cited 2026 Aug 28]. Roku Launchpad. Available from: https://developer.roku.com/en-ca/overview

34. Samsung [Internet]. [cited 2026 Aug 28]. Smart TV | Samsung One UI Tizen | Samsung Canada. Available from: https://www.samsung.com/ca/tvs/smart-tv/smart-hub-and-apps/

35. The Linux Foundation [Internet]. 2026 [cited 2026 Aug 28]. My Life as a Linux Kernel Developer and Maintainer with Paul E. McKenney. Available from: https://zoom.platform.linuxfoundation.org/webinar/register?project=tlf&id=7Gx04vvRIXFiYZJpqt1GptOUwZ5Y5cVU51xFqww%2BnROWllcfO8Ext3wQs9wpws0gI1OxNMTcyAZk%2BZXuaEk%3D&logo=thelinuxfoundation-color.svg&zoomID=96171875049

36. Husain A, Kampshoff P. McKinsey & Company [Internet]. 2026 [cited 2026 Sep 2]. The North American automotive industry’s road to resilience. Available from: https://www.mckinsey.com/industries/automotive-and-assembly/our-insights/the-north-american-automotive-industrys-road-to-resilience

37. Automotive Grade Linux’s 2025 Year in Review. Automotive Grade Linux [Internet]. 2026 Feb 19 [cited 2026 Sep 2]. Available from: https://www.automotivelinux.org/blog/automotive-grade-linuxs-2025-year-in-review/

38. Rowe R. Linux Invades Hollywood | Computer Graphics World. CGW [Internet]. 2001 Sep [cited 2026 Aug 28];24(9). Available from: https://www.cgw.com/Publications/CGW/2001/Volume-24-Issue-9-September-2001-/Linux-Invades-Hollywood.aspx

39. Robertson B. Open Source in Entertainment: How the Academy Software Foundation Creates Shared Value [Internet]. 2022 [cited 2026 Aug 28]. Available from: https://www.linuxfoundation.org/research/open-source-in-entertainment

40. Conway A. XDA [Internet]. 2026 [cited 2026 Aug 28]. Valve isn’t ditching Windows or x86, but it’s quietly making both optional. Available from: https://www.xda-developers.com/valve-isnt-ditching-windows-x86-quietly-making-optional/

41. Gosselin A. Linux Journal [Internet]. 2021 [cited 2026 Aug 28]. Linux in Healthcare - Cutting Costs & Adding Safety | Linux Journal. Available from: https://www.linuxjournal.com/content/linux-healthcare-cutting-costs-adding-safety

42. Stewart K, Smith J. Analyzing Open Source Interactions in Linux Based Medical Devices [ELISA] [Internet]. 2022 Jan 12 [cited 2026 Aug 28]. Available from: https://elisa.tech/blog/2022/01/12/analyzing-open-source-interactions-in-linux-based-medical-devices/

43. Muschelli J, Sweeney E, Crainiceanu CM. freesurfer: Connecting the Freesurfer software with R [Internet]. F1000Research; 2018 [cited 2026 Aug 28]. Available from: https://f1000research.com/articles/7-599 doi:10.12688/f1000research.14361.1

44. Harkey T, Baker D, Hagen J, Scott H, Palys V. Practical methods for segmentation and calculation of brain volume and intracranial volume: a guide and comparison. Quant Imaging Med Surg. 2022 Jul;12(7):3748761–3743761. doi:10.21037/qims-21-958

45. Hubbard T, Barker D, Birney E, Cameron G, Chen Y, Clark L, et al. The Ensembl genome database project. Nucleic Acids Res. 2002 Jan 1;30(1):38–41. doi:10.1093/nar/30.1.38

46. Moxa [Internet]. [cited 2026 Aug 28]. MIL Secured Edge and Built a Trusted System. Available from: https://www.moxa.com/en/spotlight/industrial-computing/moxa-industrial-linux/index

47. Toshiba [Internet]. [cited 2026 Aug 28]. (Part 1) Technical challenges of applying Linux to industrial equipment and Toshiba’s longstanding efforts. Available from: https://www.global.toshiba/jp/company/digitalsolution/articles/tsoul/tech/t0901.html

48. NASA Jet Propulsion Laboratory (JPL) [Internet]. [cited 2026 Aug 28]. Meet the Open Source Software Powering Nasas Ingenuity Mars Helicopter. Available from: https://www.jpl.nasa.gov/news/meet-the-open-source-software-powering-nasas-ingenuity-mars-helicopter/

49. Projects - F Prime [Internet]. [cited 2026 Aug 28]. Available from: https://fprime.jpl.nasa.gov/overview/projects/

50. Brown E. LinuxGizmos.com [Internet]. 2021 [cited 2026 Aug 28]. NASA’s Martian helicopter runs Linux. Available from: https://linuxgizmos.com/nasas-martian-helicopter-runs-linux/

51. Cohen N, Phys.org. Phys.org [Internet]. 2013 [cited 2026 Aug 28]. International Space Station making laptop migration from

Windows XP to Debian 6. Available from: https://phys.org/news/2013-05-international-space-station-laptop-migration.html

52. Vaughan-Nichols S. ZDNet [Internet]. 2020 [cited 2026 Aug 28]. From Earth to orbit with Linux and SpaceX. Available from: https://www.zdnet.com/article/from-earth-to-orbit-with-linux-and-spacex/

53. Bielik J. The real story behind SpaceX's 32,000 Linux computers in orbit. Webiano Digital & Marketing Agency [Internet]. 2026 Apr 27 [cited 2026 Aug 28]. Available from: https://webiano.digital/the-real-story-behind-spacexs-32000-linux-computers-in-orbit/

54. Edge J. Introducing Space Grade Linux. LWN.net [Internet]. 2025 Sep 9 [cited 2026 Aug 28]. Available from: https://lwn.net/Articles/1036168/

55. The Linux Foundation [Internet]. [cited 2026 Aug 28]. My Life as a Linux Kernel Developer and Maintainer with Julia Lawall. Available from: https://zoom.platform.linuxfoundation.org/webinar/register?project=tlf&id=FDTYryEwaGvbVHCTY8gmh9iFqvp0eimdJC%2BXVkctFwuwYAInlUYj3MF0xU4kOZ5XXzpqqq%2BHOoW%2BVNRym9M%3D&logo=thelinuxfoundation-color.svg&zoomID=92853776811

56. Vaughan-Nichols S. ZDNET [Internet]. [cited 2026 Aug 28]. Bob Young talks about the origins of Red Hat. Available from: https://www.zdnet.com/article/bob-young-talks-about-the-origins-of-red-hat/

57. IBM Developer [Internet]. [cited 2026 Aug 28]. Get involved with open source projects - Call for Code. Available from: https://developer.ibm.com/callforcode/solutions/projects/

58. marcin.tomaszek. Linux Kernel: History, Applications, and Major Distributions | Storware BLOG. Storware [Internet]. 2022 Oct 13 [cited 2026 Aug 28]. Available from: https://storware.eu/blog/linux-kernel-history-applications-and-major-distributions/

59. Hayward D. TechRadar [Internet]. 2012 [cited 2026 Aug 28]. The history of Linux: how time has shaped the penguin. Available from: https://www.techradar.com/news/software/operating-systems/the-history-of-linux-how-time-has-shaped-the-penguin-1113914/3

60. Rootstack [Internet]. [cited 2026 Aug 28]. Red Hat: We explain everything about its cloud solutions. Available from: https://rootstack.com/en/blog/red-hat-cloud-solutions

61. Usage Statistics and Market Share of Unix for Websites, August 2026 [Internet]. [cited 2026 Aug 28]. Available from: https://w3techs.com/technologies/details/os-unix

62. Strauss D. Linux Journal [Internet]. 1998 [cited 2026 Aug 28]. Linux Helps Bring Titanic to Life | Linux Journal. Available from: https://www.linuxjournal.com/article/2494

63. MIT Media [Internet]. 2001 [cited 2026 Aug 28]. Linux takes Hollywood by storm. Available from: https://www.media.mit.edu/broadercasting/sig/archive/5-14-2001/story5.html

64. Red Hat [Internet]. [cited 2026 Aug 28]. Volkswagen supports computer-aided engineering with a modern operating system. Available from: https://www.redhat.com/en/resources/volkswagen-research-deveopment-case-study

65. TheirStack.com [Internet]. [cited 2026 Aug 28]. Companies that use Red Hat Enterprise Linux (RHEL) (8,803). Available from: https://theirstack.com/en/technology/red-hat-enterprise-linux

66. Corbet J. Clockevents and dyntick. LWN.net [Internet]. 2007 Feb 21 [cited 2026 Aug 28]. Available from: https://lwn.net/Articles/223185/

67. StatCounter Global Stats [Internet]. [cited 2026 Aug 28]. Desktop Operating System Market Share Worldwide. Available from: https://gs.statcounter.com/os-market-share/desktop/worldwide/

68. Paraschiv L. Chrome Os - everything you need to know. FOTC [Internet]. 2022 Sep 26 [cited 2026 Aug 28]. Available from: https://fotc.com/blog/chrome-os/

69. StatCounter Global Stats [Internet]. [cited 2026 Aug 28]. Mobile Operating System Market Share Worldwide. Available from: https://gs.statcounter.com/os-market-share/mobile/worldwide/

70. The 20th Anniversary of Linux. IWB Blog [Internet]. 2011 Aug 29 [cited 2026 Aug 28]. Available from: https://blog.irvingwb.com/blog/2011/08/the-20th-anniversary-of-linux.html

71. The Linux Foundation [Internet]. [cited 2026 Aug 28]. My Life as a Linux Kernel Developer and Maintainer with Shuah Khan. Available from: https://zoom.platform.linuxfoundation.org/webinar/register?project=tlf&id=kPXt5fjBXPqzGXz3OYVXe0HYE6WpzZ1rCNROoWqXxC70EYWJduoqGBVmeMNB7h6gTitSD%2BBYtcsQjkfIhLk%3D&logo=thelinuxfoundation-color.svg&zoomID=99495497158

72. Susnjara S. The History of Kubernetes | IBM [Internet]. 2024 [cited 2026 Aug 28]. Available from: https://www.ibm.com/think/topics/kubernetes-history

73. Kienlt. Containers Are Just Linux Primitives: Namespaces, Cgroups, OverlayFS, and Podman. Medium [Internet]. 2026 Mar 25 [cited 2026 Aug 28]. Available from: https://medium.com/@kienlt.qn/containers-are-just-linux-primitives-namespaces-cgroups-overlayfs-and-podman-4f61d6c78fbc

74. Brown N. Adventures in a Cloud Native Landscape [Internet]. 2014 [cited 2026 Aug 28]. Introducing Namespaces. Available from: https://windsock.io/introducing-the-namespaces/

75. Corbet J. Notes from a container. LWN.net [Internet]. 2007 Oct 29 [cited 2026 Aug 28]. Available from: https://lwn.net/Articles/256389/

76. Larabel M. Phoronix [Internet]. [cited 2026 Aug 28]. OverlayFS Proposed For The Linux 3.18 Kernel. Available from: https://www.phoronix.com/news/MTc5OTc

77. torvalds. GitHub [Internet]. [cited 2026 Aug 28]. overlay filesystem · torvalds/linux@e9be9d5. Available from: https://github.com/torvalds/linux/commit/e9be9d5e76e34872f0c37d72e25bc27fe9e2c54c

78. Jaimes R. Linux Statistics 2026: Operating System Usage And Developer Adoption Rate. Quantumrun Foresight [Internet]. 2026 May 29 [cited 2026 Aug 28]. Available from: https://www.quantumrun.com/consulting/linux-statistics/

79. Red Hat Enterprise Linux Revenue Statistics [Internet]. 2025 Dec 23 [cited 2026 Aug 28]. Available from: https://commandlinux.com/statistics/red-hat-enterprise-linux-revenue-statistics/

80. The State of eBPF [Internet]. The Linux Foundation; 2024 [cited 2026 Aug 28]. Available from: https://ebpf.foundation/report-the-state-of-ebpf/

81. The Hybrid Cloud Mainframe: Opening up to cloud-native developers [Internet]. 2025 [cited 2026 Aug 28]. Available from: https://openmainframeproject.org/wp-content/uploads/sites/14/2025/10/Paper_Hybrid_Mainframe-1.pdf

82. Corbet J. The realtime preemption pull request. LWN.net [Internet]. 2024 Sep 20 [cited 2026 Sep 10]. Available from: https://lwn.net/Articles/990985/

83. Corbet J. Approaches to realtime Linux. LWN.net [Internet]. 2004 Oct 12 [cited 2026 Sep 10]. Available from: https://lwn.net/Articles/106010/

84. Kernel index [LWN.net] [Internet]. [cited 2026 Sep 10]. Available from: https://lwn.net/Kernel/Index/#Realtime

85. Corbet J. The realtime preemption end game — for real this time. LWN.net [Internet]. 2024 Sep 6 [cited 2026 Sep 10]. Available from: https://lwn.net/Articles/989212/

86. Real-Time Linux. Realtime Linux [Internet]. 2026 Sep 10 [cited 2026 Sep 10]. Available from: https://realtime-linux.org/

87. linux-kernel - [patch 1/1] MAINTAINERS: Add co-maintainers for time[rs] [Internet]. [cited 2026 Sep 10]. Available from: https://lists.openwall.net/linux-kernel/2024/03/25/4185

88. The Linux Associates [Internet]. [cited 2026 Aug 28]. The Future of Edge Computing with Linux: Powering the Next Wave of Innovation. Available from: https://thelinuxassociates.com/blog/the-future-of-edge-computing-with-linux/

89. Linux In Edge Computing Deployments Statistics 2026. Command Linux [Internet]. 2026 Mar 17 [cited 2026 Aug 28]. Available from: https://commandlinux.com/statistics/linux-in-edge-computing-deployments/

90. Whittaker G. Linux Journal [Internet]. 2024 [cited 2026 Aug 28]. Linux in the Edge Computing Ecosystem and IoT Gateway Technologies. Available from: https://www.linuxjournal.com/content/linux-edge-computing-ecosystem-and-iot-gateway-technologies

91. Ubuntu Core: The imm [Internet]. [cited 2026 Aug 28]. Ubuntu Core. Available from: https://ubuntu.com/core

92. Lavie D. Datadog [Internet]. 2023 [cited 2026 Aug 28]. A gentle introduction to XDP. Available from: https://www.datadoghq.com/blog/xdp-intro/

93. The Linux Foundation [Internet]. [cited 2026 Aug 28]. My Life as a Linux Kernel Developer and Maintainer with Mark Brown. Available from: https://zoom.platform.linuxfoundation.org/webinar/register?project=tlf&id=5wh9bf%2FuqZKOQz6ilOyhFJbFgl1AnSTZqDqZj842%2FdDGpKQaAsUyOC66%2BlRbg6LfLFyBwOx2h66rAcu9XGY%3D&logo=thelinuxfoundation-color.svg&zoomID=97172691626

94. Corbet J. Improved load balancing with machine learning. LWN.net [Internet]. 2025 Jul 1 [cited 2026 Aug 28]. Available from: https://lwn.net/Articles/1027096/

95. The Linux Foundation. Mentorship Session: Designing Custom Linux Schedulers with sched_ext [Internet]. 2025 [cited 2026 Aug 28]. Available from: https://www.youtube.com/watch?v=jsLjg9tGuVI

96. Zheng Y, Fan J, Fu Q, Yang Y, Zhang W, Quinn A. AgentCgroup: Understanding and Controlling OS Resources of AI Agents [Internet]. arXiv; 2026 [cited 2026 Aug 28]. Available from: http://arxiv.org/abs/2602.09345 doi:10.48550/arXiv.2602.09345

97. eunomia-bpf/agentcgroup [Python] [Internet]. eunomia-bpf; 2026 [cited 2026 Aug 28]. Available from: https://github.com/eunomia-bpf/agentcgroup

98. Alvarez E, Klieger B, Steele G. NVIDIA [Internet]. 2026 [cited 2026 Aug 28]. Building for the Rising Complexity of Agentic Systems with Extreme Co-Design. Available from: https://developer.nvidia.com/blog/building-for-the-rising-complexity-of-agentic-systems-with-extreme-co-design/

99. Hardware-Software Codesign - an overview [Internet]. [cited 2026 Aug 28]. Available from: https://www.sciencedirect.com/topics/engineering/hardware-software-codesign

100. Global Market Insights Inc. [Internet]. 2026 [cited 2026 Aug 28]. RISC-V Market Size & Share, Forecasts Report 2026-2035. Available from: https://www.gminsights.com/industry-analysis/risc-v-market

101. AI Coding Assistants [Internet]. [cited 2026 Aug 28]. Available from: https://docs.kernel.org/process/coding-assistants.html

102. Sharwood S. theregister [Internet]. 2026 [cited 2026 Aug 28]. Linus Torvalds to ‘start being more hardnosed’ about ‘pointless pull requests’ – some of which come from AIs. Available from: https://www.theregister.com/oses/2026/05/25/linus-torvalds-to-start-being-more-hardnosed-about-pointless-pull-requests-some-of-which-come-from-ais/5245549

103. sashiko-dev/sashiko [Rust] [Internet]. sashiko.dev; 2026 [cited 2026 Aug 28]. Available from: https://github.com/sashiko-dev/sashiko

104. Re: Linking Patchwork with Sashiko? - Linus Torvalds [Internet]. [cited 2026 Aug 28]. Available from:

https://lore.kernel.org/linux-media/CAHk-=wi4zC+Ze8e+p3tMv8TtG_80KzsZ1syL9anBtmEh5Z40vg@mail.gmail.com/

105. Ahmad M. The Conversation [Internet]. 2026 [cited 2026 Aug 28]. Mythos AI is a cybersecurity threat, but it doesn’t rewrite the rules of the game. Available from: https://theconversation.com/mythos-ai-is-a-cybersecurity-threat-but-it-doesnt-rewrite-the-rules-of-the-game-281268 doi:10.64628/AAI.476td45um

106. IBM Newsroom [Internet]. [cited 2026 Aug 28]. IBM and Red Hat Commit $5 Billion to Redefine the Future of Open Source in the AI Era. Available from: https://newsroom.ibm.com/2026-05-28-ibm-and-red-hat-commit-5-billion-to-redefine-the-future-of-open-source-in-the-ai-era

107. Risi C. InfoQ [Internet]. [cited 2026 Aug 28]. Linux Foundation Launches Akrites to Protect Critical Open Source Software from AI-Powered Threats. Available from: https://www.infoq.com/news/2026/07/akrites-open-source-ai-threats/

108. The Linux Foundation [Internet]. [cited 2026 Aug 28]. My Life as a Linux Kernel Developer and Maintainer with Sasha Levin. Available from: https://zoom.platform.linuxfoundation.org/webinar/register?project=tlf&id=r8dhKESMsJLWbcQBAtPD%2Blu6PLH%2BOmyUOlnuQ9a0RaeLYfaZkG7MlgK6t7IKVQtuk5FTB29GmWXf%2F5m2j0Y%3D&logo=thelinuxfoundation-color.svg&zoomID=98005374249

109. Vaughan-Nichols S. ZDNET [Internet]. [cited 2026 Aug 28]. “Rust makes coding fun again”: Why Linux is moving away from C, according to Greg Kroah-Hartman. Available from: https://www.zdnet.com/article/greg-kroah-hartman-linux-kernel-rust/

110. Rustify [Internet]. 2026 [cited 2026 Aug 28]. Rust in the Linux Kernel: What It Means in 2026. Available from: https://rustify.rs/articles/rust-in-linux-kernel-2026

111. Vaughan-Nichols SJ. Rust Integration in Linux Kernel Faces Challenges but Shows Progress. The New Stack [Internet]. 2025 Feb 19 [cited 2026 Aug 28]. Available from: https://thenewstack.io/rust-integration-in-linux-kernel-faces-challenges-but-shows-progress/

112. DomainsProject.org. Rusted Kernel [Internet]. [cited 2026 Aug 28]. Rust in the Linux Kernel — a version-by-version analysis. Available from: https://rusted-kernel.com/

113. Corbet J. The (successful) end of the kernel Rust experiment. LWN.net [Internet]. 2025 Dec 10 [cited 2026 Aug 28]. Available from: https://lwn.net/Articles/1049831/

114. ossf/best-practices-badge [Ruby] [Internet]. Open Source Security Foundation (OpenSSF); 2026 [cited 2026 Aug 28]. Available from: https://github.com/ossf/best-practices-badge

115. Wondrasek JA. The XZ Utils Backdoor CVE-2024-3094 and the Multi-Year Social Engineering Campaign Behind It. SoftwareSeni [Internet]. 2026 Feb 15 [cited 2026 Aug 28]. Available from: https://www.softwareseni.com/the-xz-utils-backdoor-cve-2024-3094-and-the-multi-year-social-engineering-campaign-behind-it/

116. Humble C. The New Stack [Internet]. [cited 2026 Aug 28]. The XZ Utils aftermath: Inside the mission to stop the next global

backdoor before it starts. Available from: https://thenewstack.io/commonhaus-open-source-governance/

117. Linus Torvalds speaks on AI, More Adobe on Linux, big browser redesigns [Internet]. 2026 [cited 2026 Aug 28]. (Linux & Open Source News). Available from: https://podcast.thelinuxexp.com/@tlenewspodcast/episodes/linus-torvalds-speaks-on-ai-more-adobe-on-linux-big-browser-redesigns

118. Interview with Shuah Khan [Zoom]. 2026.

119. Corbet J. Development statistics for the 7.0 kernel. LWN.net [Internet]. 2026 Apr 13 [cited 2026 Aug 28]. Available from: https://lwn.net/Articles/1066723/

120. Corbet J. Development statistics for the 7.1 kernel. LWN.net [Internet]. 2026 Jun 15 [cited 2026 Aug 28]. Available from: https://lwn.net/Articles/1077425/

121. Vaughan-Nichols S. ZDNET [Internet]. 2026 [cited 2026 Aug 28]. Linux after Linus? The kernel community finally drafts a plan for replacing Torvalds. Available from: https://www.zdnet.com/article/linux-community-project-continuity-plan-for-replacing-linus-torvalds/

122. Perlow J. Mentorship in Open Source: Exploring the Intrinsic, Economic, and Career Value of Mentorship Programs [Internet]. 2023 [cited 2026 Aug 28]. Available from: https://www.linuxfoundation.org/research/mentorship-in-open-source

123. Mentorship [Internet]. [cited 2026 Aug 28]. Available from: https://mentorship.lfx.linuxfoundation.org/#projects_all

124. LF Events [Internet]. [cited 2026 Aug 28]. LF Live: Mentorship Series. Available from: https://events.linuxfoundation.org/lf-live-mentorship-series/

125. The Linux Foundation [Internet]. 2021 [cited 2026 Aug 28]. Interview with Shuah Khan, Kernel Maintainer & Linux Fellow - Linux Foundation. Available from: https://www.linuxfoundation.org/blog/blog/interview-with-shuah-khan-kernel-maintainer-linux-fellow

126. LFX [Internet]. [cited 2026 Aug 28]. Mentorship Dashboard. Available from: https://metabase.tools.lfx.dev/dashboard/63-mentorship-dashboard?daterange=&gender=&parent_project=&persona=mentee&program_name=&project_name=&status=&term=

127. The Linux Foundation [Internet]. 2026 [cited 2026 Aug 28]. LF Live Maintainer Series: My Life as a Linux Kernel Developer and Maintainer with Greg KH and Shuah Khan. Available from: https://www.linuxfoundation.org/webinars/lf-live-maintainer-series-my-life-as-a-linux-kernel-developer-and-maintainer-with-greg-kh-and-shuah-khan?hsLang=en

128. PIB [Internet]. 2025 [cited 2026 Aug 28]. India’s UPI Revolution: Over 18 billion Transactions Every Month, A Global Leader in Fast Payments. Available from: https://static.pib.gov.in/WriteReadData/specificdocs/documents/2025/jul/doc2025720589601.pdf

THE LINUX FOUNDATION | Research

2021 年に設立された **Linux Foundation Research** は、オープンソースにおけるコラボレーションの規模拡大を探り、新たな技術動向、ベスト プラクティス、そしてオープンソース プロジェクトが世界に与える影響についての洞察を提供します。Linux Foundation Research は、プロジェクトのデータベースやネットワークを活用し、定量的および定性的な手法におけるベスト プラクティスに尽力することで、世界中の組織にとって有益なオープンソースに関する知見を得るための頼れるライブラリを構築しています。



Copyright © 2026 **The Linux Foundation**

本レポートは、**Creative Commons Attribution-NoDerivatives 4.0 International Public License** に基づいてライセンスされています。

本著作物を引用する場合は、以下の形式で引用してください：Anna Hermansen, Hilary Carter, and Daniel German, “35 Years of Linux: The Open Source Engine of Global Infrastructure and Innovation,” foreword by Dirk Hohndel, The Linux Foundation, October 2026.

facebook.com/TheLinuxFoundation

x.com/linuxfoundation

linkedin.com/company/TheLinuxFoundation