



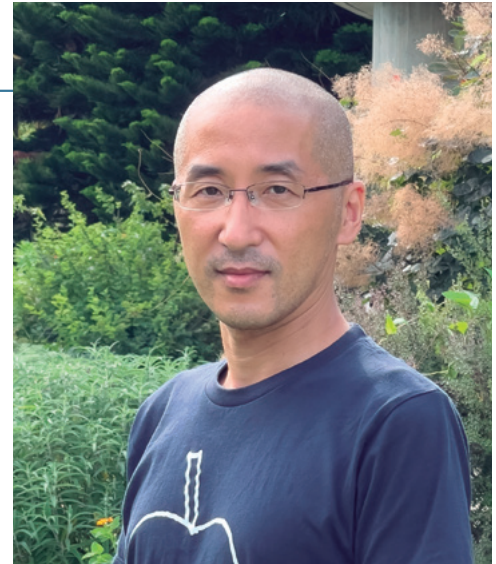
NTTソフトウェアイノベーションセンタ  
特別研究員

**藤田 智成** Tomonori Fujita

## 「高信頼なシステムソフトウェア技術」で新たなエコシステムを創出し、グローバルに貢献

インターネットが普及しICTが生活に溶け込んだ現代では、ソフトウェアの信頼性への要求が非常に高まっています。高い性能が求められる分野のソフトウェア開発で現在主流のC言語は、柔軟で自由なハードウェア操作が可能である一方で、開発者の単純な誤りが不具合やセキュリティリスクを招く点が懸念されています。そのため高い性能と現代の要求に合った信頼性を満たすソフトウェア開発のためのプログラミング技術が求められており、C言語に代わる新たなプログラミング言語の取り組みも進んでいます。今回は「高信頼なシステムソフトウェア技術」の研究に取り組む、藤田智成特別研究員にお話を聞きました。

◆PROFILE：2000年早稲田大学大学院理工学研究科修士課程修了。同年、日本電信電話株式会社入社。2015年より特別研究員。オペレーティングシステムをはじめとしたシステムソフトウェアの研究開発に従事。2014年10月日本電信電話株式会社社長表彰、2015年2月情報処理学会ソフトウェアジャパンワード等を受賞。



### 「高性能」「安全性」を両立したRustで新たなソフトウェアを実現

■「高信頼なシステムソフトウェア技術」では、どのような取り組みをされているのでしょうか。

私が研究している「高信頼なシステムソフトウェア技術」とは、ハードウェアの制御を行うシステムソフトウェアがハードウェアの障害やネットワークからの攻撃などに耐え、想定どおりに動作し続けることをめざす技術です。これを達成するにはさまざまな基本技術が必要になり、現在はシステムの停止や外部からの攻撃につながる、ソフトウェアの問題の回避に取り組んでいます。具体的には、システムソフトウェア開発のプログラミング言語として、主流のC言語ではなく「Rust」という開発者の誤りを防ぐための機能（安全性）を備えたプログラミング言語を用いて、高性能なシステムソフトウェアの実現をめざしています。

この研究の背景として、インターネットが普及した現代でソフトウェアの信頼性への要求が高まったことが挙げられます。インターネットが一般に普及し始めた1990年代は、その利用者数も限られていたうえ、用途も生活に影響を与えないものばかりでした。しかし、現在では社会全体でICT活用が活発になり、インターネットが生活インフラに組み込まれ、システムの停止が許容されない状況となっています。さらにクレジットカード番号のような重要な情報をインターネットでやり取りしているため、システムに不備があった場合に金銭的な被害の発生などのセキュリティ事故が

起きてしまいます。そのようなトラブルが発生することなく、社会インフラとしてのインターネットが常に想定どおりに動作し続けることが当然とみなされており、インターネットを支えるシステムソフトウェアの信頼性への期待値も非常に上がっています。しかし、信頼性の期待値が大きく変化したにもかかわらず、現在のインターネットを支えるために1990年代に設計されたシステムソフトウェアが数多く使われているという点が大きな問題となっています。

信頼性の高いシステムを実現するためには、ソフトウェアの不具合と外部攻撃の原因となる脆弱性を防ぐ必要があり、プログラミング言語の選択は非常に重要です。現在主流となっているC言語は、開発者がメモリやCPUといったハードウェアに対する柔軟で自由な操作ができるため、ハードウェアの性能を引き出しやすい言語です。しかしC言語は1970年代に開発されたプログラミング言語で、安全性、すなわち不具合や脆弱性を起こす誤りを防ぐために開発者を支援する仕組みは考慮されておらず、リスクが高いことが知られています。2022年にアメリカ国家安全保障局(NSA)は、C言語以外のプログラミング言語を使うことを推奨するガイダンスを発表しています。2000年ごろから、高性能なソフトウェアをターゲットとする安全性を備えたプログラミング言語が数多く提案されていますが、現在ではその中の1つであるRustを用いて、インターネットを支える新しいシステムソフトウェア実現に向けた研究に取り組んでいます。

### ■プログラミング言語「Rust」の特徴について教えてください。

Rustは「高性能のソフトウェアを安全に実装できる」というのが大きな特徴です。例えばJavaやGo言語のメモリ管理方法は、ソフトウェアの実行中にメモリの使用状況を把握し、使われなくなったメモリ領域を解放する「ガベージコレクション」と呼ばれる仕組みにより不正なメモリ操作を防ぎます。このガベージコレクションは安全なメモリ操作を実現する一方で、ソフトウェアの実行中に動作する仕組みであるため、性能に悪影響を及ぼします。そこでRustでは性能を低下させるガベージコレクションを使わず、ソフトウェアのコンパイル（ソースコード変換）時にメモリの使用状況を割り出し、適切に管理できるように設計されています。「高性能」と「安全性」の両立は難しい問題ですが、Rustは開発者に特有の設計や表現を強制することで、両方を実現しています。

現在私は、オペレーティングシステムとBGP（Border Gateway Protocol）デーモン\*の実現に取り組んでいます（図1）。オペレーティングシステムに関しては、MicrosoftやGoogleなどの他社の開発者と連携して、Linuxオペレーティングシステムの中核機能（カーネル）をRustで実装することに取り組んでいます。Linuxカーネルのソフトウェア規模や普及状況を考えて、Rustで実装した新たなオペレーティングシステムでLinuxを置き換えることは現実的ではないと考え、C言語で実装されているLinuxカーネルを少しずつRustで置き換えていくことをめざしています。

またインターネットの基幹であるBGPプロトコルをサポートするBGPデーモンは、大規模なクラウドインフラのネットワーク内でも使われるなど、高い性能と信頼性が求められるシステムソフトウェアの1つです。現在C言語で実装されたBGPデーモンが広く使われていますが、複数のCPUを活用するための並列性が低く、最新のハードウェアの性能を引き出せていません。そこでRustが備える並列性を安全に扱う支援を活用し、新たなBGPデーモンを実装したところ、広く使われている実装と比較して5倍以上の性能を実現することができています。

### ■現在ご研究で苦労されている点について教えてください。

新しいプログラミング技術が広く使われるようになるまでには時間がかかります。特にインフラの基幹となるシステムソフトウェア分野は新しい技術を取り入れるのに時間がかかる傾向があります。

これまでC言語に代わる安全性を備えたプログラミング言語が数多く開発されましたが、システムソフトウェア開発ではいまだにC言語が主流です。その理由の1つは、開発者にとって新しい

\* デーモン：Linux等のUNIX系OSでメインメモリ上に常駐して特定の機能（利用者の操作とは無関係に処理を行うバックグラウンドプロセス等）を提供するプログラム。

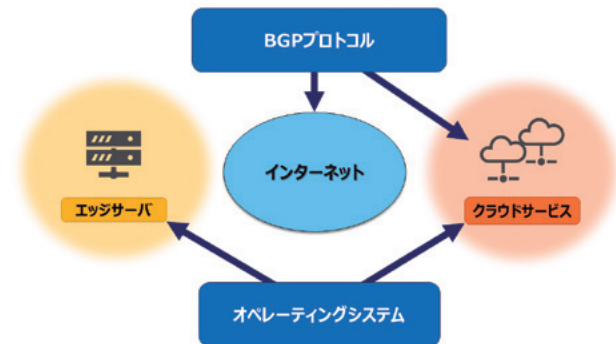


図1 現在ターゲットとしているシステムソフトウェア

言語でソフトウェアを開発・保守することが大きな負担であるためだと思われます。

またRustの場合は、従来の言語にみられない独自の仕様が普及の障壁になっているといわれています。実際にLinuxカーネルの機能をC言語に加えて、Rustでも実装できるようにする際も賛否両論がありました。議論を経て、Linuxカーネルの機能をRustで実装できるようになりましたが、今のところLinuxカーネルにはRustで実装された機能がありません。

こうした現状を打破する近道はないと思い、まずは「小さな機能から実装して有用性を示す」ことに気を付けています。具体的に、Linuxカーネルのネットワーク通信のハードウェアを制御するデバイスドライバをRustで実装できるようにする機能を提案したところ、開発者から大きな抵抗がありました。そこで影響範囲を小さくして、ネットワークデバイスドライバの一部機能をRustで実装できるようにすることを交渉しています。

オープンソースソフトウェア（OSS）の開発で自分の提案が100%受け入れられることはまれで、私自身も若いころは提案が断られるたびにショックを受けていました。しかしOSSの開発に長年携わる中で、たとえ提案が10%しか通らなかったとしても、他の開発者の意見を取り入れながら提案を改善して進み続ける重要性に気が付きました。

## 多くの人を巻き込んで世の中に大きなインパクトを与える研究に取り組む

### ■今後の研究ビジョンについて教えてください。

次のステップとして、Rustのような安全性を備えたプログラミング言語をシステムソフトウェアに適用する価値を示す必要があると考えています。Linuxカーネルの開発コミュニティでも賛否両論あったように、「システムソフトウェアにおいてC言語以外を使うことに十分なメリットがあるか」という問いにはまだ結論が出ていないようです。LinuxカーネルでRustの利用例を増やすためには、Rustで実装された広く使われる機能を採用してもらい、

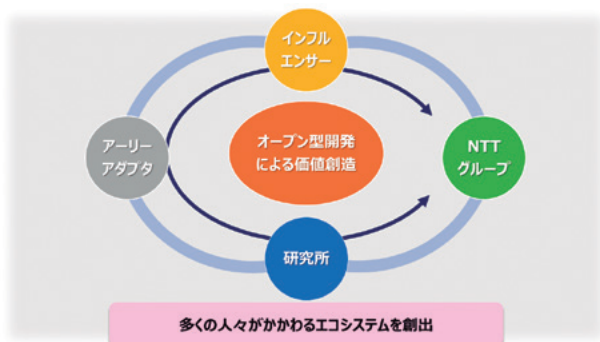


図2 オープン型開発による価値創造のビジョン

不具合や脆弱性に関してメリットを示す必要があるように思います。また安全性を備えたプログラミング言語を使うことですべての不具合や脆弱性を防ぐことはできませんので、例えば設計の不具合を防ぐなど、他のアプローチでも検討を進める予定です。

システムソフトウェアの不具合や脆弱性を防ぐための研究成果をデファクトの技術として普及させることで、業界全体のシステムソフトウェアの信頼性を高めることをめざしています。それによって私たちの生活を支えるインフラの信頼性が向上するだけでなく、従来不具合や脆弱性の対策に費やされてきたリソースが新しい機能開発に使われ、サービスの価値向上につながると考えています。

またNTTが提唱するIOWN (Innovative Optical and Wireless Network) 構想において、システムの停止や不正な利用を避けるための技術は非常に重要なものです。IOWNでこれまで以上に生活がデジタルと融合してICTが不可欠になる世界観において、もし「サービスが停止しました」という事態が起きてしまうと利用者の不満はこれまで以上に大きなものになります。そのような事態を回避して安全で快適な社会を実現することをめざして、今後もOSSなどのオープンな開発手法でNTT以外にもさまざまな人々がかかわるエコシステムを創出し、グローバルに新たな価値を創造していきます (図2)。

■最後に研究者・学生・ビジネスパートナーの方々へメッセージをお願いします。

私の研究モットーは「多くの人に使われて、人の生活をさらに良くする技術をつくる」ことです。NTTに入社して間もないころに、上司から「研究成果が商用化された場合に、その利益で何人が生活できるのか」と研究テーマについて話をされたのを今でも覚えています。数10名程度の生活しか支えられずグループ内しか使われない技術をめざすのではなく、「世の中をもっと大きく変えて何万人にも影響するテーマを考えるように」というアドバイスだったと認識しており、現在もこの考え方は大切にしています。

そして技術自体も誰かに使ってもらうことで初めて成長していくものです。多くの人に使ってもらうことで新たなマーケットができて多くの人がそれでお金を稼げるようになる、逆にお金を生み出さない技術は普及しないと考えています。

今後の研究でも新しいマーケットを創出する技術を生み出していくことが目標です。私のグループでは、NTTだけに技術をとどめておくのではなく、OSSなどのオープンな開発手法を使って他社の方と協力しながら世の中に普及させ、そしてより大きなインパクトのある技術に育てていくことでNTTの取り組みが広がるように研究を進めています。国内でNTTと他社が協業する際にはNTTは主導的な立場を取ることありますが、メジャーなグローバルOSSプロジェクトだとNTTは主要なプレイヤーではありません。そして他社の方と利害関係が完全に一致することはほぼないため、利害関係を調整して自分が向きたい方向に舵を取らなければいけません。そのため新しいコミュニティに参加するたびに、コミュニティのメンバーと直接会って話をする機会を設けたりしています。一見地道な活動ですが、やはりメールやチャットだけだとお互いに伝わらない情報でも、お互いに膝を突き合わせて会話をすることで解像度が上がるため、相手への理解を深めながらコミュニティ内の発言力や影響力を高めています。

インターネットも元々多くの人たちが集まってつくられたもので、これから私自身も世の中にインパクトを与えるために、1人研究室に閉じこもって考えるのではなく、多くの人を巻き込んで協力しながら新たな価値を創出していきます。これから一緒に社会にインパクトを与える活動にぜひ参加していただけることを願っています。



(今回はリモートにてインタビューを実施しました)