IPA

2024 年度 未踏 IT 人材発掘・育成事業 採択案件評価書

1. 担当 PM

曾川 景介（newmo 株式会社 CTO）

2. クリエータ氏名

伊組 烈火（フリーランス）

3. 委託金支払額

2,880,000 円

4. テーマ名

Capability-Based Microkernel によるセキュアなユーザレベルメモリ管理システム

5. 関連 Web サイト

- A9N Microkernel：
  https://github.com/horizon2038/A9N
- A9N Boot Protocol(x86_64) に従った Reference Bootloader 実装：
  https://github.com/horizon2038/A9NLoader
- A9N Microkernel 上で動作する OS を開発するための Rust 製 Framework：
  https://github.com/horizon2038/Nun

6. テーマ概要

本プロジェクトでは、従来のオペレーティングシステム（OS）カーネルに見られる複雑化やセキュリティリスク、保守性の課題に正面から向き合い、それらを根本的に解決するための新たなアーキテクチャとして、Capability-Based Microkernel「A9N」の開発と、それを基盤としたユーザレベルのメモリ管理システムの構築に取り組んだ。A9N は、機能を最小限に抑えた Microkernel として設計され、アクセス制御には Capability ベースのセキュリティモデルを採用することで、最小特権の原則（PoLP）を徹底し、システムの安全性と柔軟性を飛躍的に高めた。

また、A9N はプロセス間通信（IPC）の高速化にも注力し、従来の Microkernel で問題となっていた通信レイテンシの課題を解決した。世界最速クラスの IPC 性能を実現するなど、性能面でも高い水準を達成した。この Microkernel 上に

構築されたユーザレベルのオペレーティングシステム「KOITO」では、プロセス管理やメモリ管理をユーザ空間で安全に実行できる仕組みが整えられており、POSIX 互換のシステムコールも提供されている。

7. 採択理由

Capability-Based Security を採用することで、最小特権の原則に基づくセキュアなメモリ管理を実現すると同時に、ユーザレベルでのメモリ管理を行うことで、柔軟性も同時に満たす OS を作ることができると提案者は考えている。メモリを Capability によって抽象化するという試みである。故に本提案は既存の同様のアプローチに対して、メモリ管理部分を主軸に置いたアプローチをとっている。また、Container や Hypervisor など仮想化技術との親和性や、Wasm などでの活用の可能性も評価した。

8. 開発目標

本プロジェクトにおける当初の開発目標は、Capability-Based Microkernel の設計と実装を通じて、現代のオペレーティングシステムが抱える構造的な課題を根本から見直し、より安全かつ柔軟で拡張性の高いシステム基盤を構築することにあった。特に、肥大化したモノリシックカーネルに起因する保守性の低下やセキュリティリスクを解決するため、最小限の機能のみをカーネルに保持し、その他の機構はユーザ空間に分離する Microkernel アーキテクチャを採用した。その上で、あらゆるリソースや操作に対してトークンベースのアクセス制御を行う Capability-Based Security モデルを導入し、最小特権の原則（PoLP）を徹底することで、高い安全性を確保することを目指した。

あわせて、Microkernel が一般に抱える性能面の課題、特にプロセス間通信（IPC）の低速性に対しても、新たな通信機構を設計・実装することで、高速かつ低レイテンシな IPC を実現し、既存の Microkernel（seL4 や Zircon など）を上回る実行性能を獲得することを目標とした。また、仮想メモリ管理においても、ページテーブルの操作やメモリ割り当てといった機構をユーザ空間に移すことで、柔軟で安全なユーザレベルメモリ管理を可能とする新しい実装モデルを提案・実現することを計画していた。

さらに、こうした低レイヤーのシステム開発においても、モダンなプログラミング言語と設計思想を積極的に取り入れ、Rust や C++20 といった型安全で表現力の高い言語を用いることで、バグの混入を防ぎつつ高い保守性を実現することも開発目標の一つであった。特に、標準ライブラリに依存しない軽量なエラーハンドリングライブラリを自作し、組み込みやカーネルのような制約下でも安全なエラー処理が可能な環境を構築することを計画していた。

これらを総合して、本プロジェクトは、セキュアで高速な Microkernel とその上で動作するユーザ空間のプロセス管理・メモリ管理基盤を一貫して設計・構

築することを開発初期段階からの主要な開発目標として掲げていた。

9. 進捗概要

本プロジェクトでは、当初掲げていた Capability-Based Microkernel の設計と、ユーザレベルでの安全なメモリ管理システムの構築という目標に対して、計画を大きく上回る進捗を達成した。まず、Microkernel 本体である「A9N」は、Capability-Based Security を中心とした設計に基づいてフルスクラッチで開発し、プロセス管理、メモリ管理、割り込み処理、デバイスアクセスなどの基本的なカーネル機能を、高度にモジュール化された形で実装した。特にプロセス間通信（IPC）に関しては、メッセージ転送やコンテキストスイッチの最適化により、世界最速クラスの通信性能を実現することに成功した。

加えて、カーネルの移植性を高めるため、ハードウェア依存部分を抽象化する HAL（Hardware Abstraction Layer）を導入し、アーキテクチャ非依存の構造を徹底した。また、メモリ管理においては、ユーザ空間における動的なメモリ割り当てと回収を実現するため、SLAB Allocator および Buddy Allocator を組み合わせた構成を採用し、Capability の仕組みを活用して安全性と効率性を両立したユーザレベルメモリ管理機構を構築した。

これらのカーネル機能を実際に活用するための OS として、「KOITO」の開発も並行して進められ、初期プロセス（Init Server）や POSIX 互換のシステムコールサーバ、ユーザレベルメモリ管理サーバなどを含む複数のサーバ群を実装した。さらに、A9N 上で OS を構築するための Rust 製フレームワーク「Nun」も新たに開発し、安全で拡張性の高いアプリケーション開発基盤を整備した。この Nun の設計には、クリーンアーキテクチャやドメイン駆動設計（DDD）といった最新のソフトウェア設計手法が取り入れており、低レイヤーでありながら高い可読性と保守性を確保している。

また、当初の計画には含まれていなかったが、プロジェクトの中盤以降にはハイパーバイザ機能の実装にも着手し、仮想 CPU や仮想メモリ空間の管理に関する基本的な API の整備が完了している。現時点では仮想マシンモニタ（VMM）上での Linux の動作には至っていないものの、機能の多くは完成しており、今後の対応が可能な段階にある。最終的には、A9N ベースの VMM で Linux をサポートするという目標に向けて、意欲的に開発を継続している。

さらに、A9N と Nun を支える基盤ライブラリとして、C++20 を用いたエラーハンドリングライブラリ「liba9n」も独自に開発し、例外処理に依存せず、安全かつ効率的なエラーハンドリングが可能な仕組みを実現している。加えて、x86_64 上でカーネルや初期プロセスを起動するための Boot Loader「A9NLoader」も整備しており、A9N ベースのシステムを独立して起動可能な形にまとめている。このように、A9N Microkernel とそれを取り巻く関連コンポーネントは、本プロジェクト期間中において大きな技術的進展を遂げ、システ

ムソフトウェアとしての実用的な土台を築くに至った。

10. プロジェクト評価

本プロジェクトは、オペレーティングシステムの根幹技術に対する深い理解と、極めて高度な実装力によって構築された、現代的かつ先進的な Microkernel システムの開発である。開発された Capability-Based Microkernel「A9N」は、既存のモノリシックカーネルが抱える肥大化や保守性の低下、セキュリティ上のリスクといった問題を真正面から捉え、最小限の機能のみをカーネルにとどめ、その他の機能をユーザ空間に分離するという明確な方針のもとに設計されている。特に、操作ごとに明示的な権限（Capability）を必要とする設計は、最小特権の原則を徹底し、システムの安全性を高めている点で優れている。

本プロジェクトの技術的な完成度は非常に高く、プロセス間通信（IPC）においては既存の Microkernel（seL4、Zircon など）を上回る世界最速クラスの性能を達成しており、Microkernel が一般的に抱える性能面のボトルネックに対しても、実装レベルで明確な解決を示している。さらに、ユーザ空間におけるメモリ管理については、SLAB Allocator と Buddy Allocator の組み合わせによって、安全かつ効率的な動的メモリ管理を実現しており、これは Capability をベースとした設計との整合性も非常に高い。

当初の計画を大きく超えて、プロジェクト中盤からはハイパーバイザ機能の実装にも着手され、A9N 上で動作する仮想マシンモニタ（VMM）の基本的な構造も整備された。これにより、Microkernel としての役割に加えて、仮想化基盤としてのポテンシャルも大きく広がっており、今後はこの VMM 上で Linux を動作させることを視野に入れた開発が継続されている。

加えて、A9N 上でユーザレベルの OS を構築するための Rust 製フレームワーク「Nun」が新たに開発されており、こちらは設計段階からクリーンアーキテクチャやドメイン駆動設計（DDD）といったモダンなソフトウェア設計手法を取り入れている点が特徴的である。これにより、OS 開発という低レイヤーかつ複雑な領域においても、高い可読性、保守性、再利用性を実現しており、技術的洗練度の高さが際立っている。

liba9n や A9NLoader といった周辺コンポーネントの整備も進んでおり、A9N ベースのシステムを一貫して構築・起動できる統合的な仕組みが形成されている点も、プロジェクトの完成度を大きく高めている。

11. 今後の課題

既にたくさんの成果を残している本プロジェクトには様々な面で今後の発展が期待される。例えば、現時点ではアーキテクチャが x86_64 に限定されているが。RISC-V などの他のアーキテクチャへのポーティングを進めるなど、より多様なプラットフォームでの動作を実現するなどが挙げられる。また、POSIX 互

換性についても、現状では提供されているシステムコールが限定的であり、標準的な UNIX 系アプリケーションを移植可能にするためには、mlibc などのより高度な互換ライブラリへの移行を含めた対応が求められる。

さらに、現在進行中の仮想化機能についても、仮想 CPU や仮想アドレス空間といった主要なコンポーネントは概ね実装されつつあるものの、VMM もまだ完全版とは言えず、今後は Linux をゲスト OS として動作させることが目標となっている。これまでは伊組氏の圧倒的な開発力によって開発が推進されてきたが、今後はいかに外部開発者にプロダクトを届け、コミュニティを形成していくかといった運用面での課題が重要な検討事項となる。しかし、さまざまなテクノロジーが加速度的に進化する現代において、一人の人間が持ちうる影響力もかつてないほどに広がっている。その中で、どこまで伊組氏が一人の力で突き抜けることができるのか、このプロジェクトはその可能性に真正面から挑んでおり、今後の展開が純粋に楽しみでならない。