

## スクリプト言語 Konoha によるカーネル診断スクリプトの開発

内田 篤史<sup>†</sup> 井出 真広<sup>†</sup>  
菅谷 みどり<sup>†</sup> 倉光 君郎<sup>†,☆</sup>

近年、スクリプト言語を用いたソフトウェア開発が広がっている。しかし、実行パフォーマンスがクリティカルな分野への応用は限定的であり、特に組み込み開発の分野においての利用はなされてこなかった。我々は静的型付けスクリプティング言語 Konoha を使い、言語処理系をカーネルへ移植し、スクリプティング言語を用いてカーネル内のモニタリングツールの開発を行う。本発表では Konoha 言語のカーネル内での利用法について報告し、モニタリングツールとしての適用の可能性について議論する。

### A kernel diagnostic script with statically scripting language Konoha

ATSUSHI UCHIDA,<sup>†</sup> MASAHIRO IDE,<sup>†</sup> MIDORI SUGAYA<sup>†</sup>  
and KIMIO KURAMITSU<sup>†,☆</sup>

Recently, software developers often use a scripting language. However, application execution performance critical areas is limited, the use of in the field of embedded development have not been particularly made. We use a statically-typed scripting language Konoha, ported to the kernel language processing system, the development of a monitoring tool in the kernel using a scripting language. In this paper, we report on the usage of Konoha scripting language in the Linux kernel to discuss the possibility of the application as a monitoring tool.

#### 1. はじめに

スクリプト言語は、プログラミングのしやすさや生産性の高さなどから近年ソフトウェアの開発で広く利用されている。しかし、実行時のパフォーマンスが重要である分野での利用は限定的であり、特に組み込み開発の分野での利用はなされてこなかった。オペレーティングシステム (OS) のカーネルは、実行パフォーマンスが強く要求される代表的なソフトウェアであり、Linux カーネルは主に C 言語によって開発されている。カーネルの開発においてデータを表現する手段として構造体を多用している。構造体はデータに関連する関数間で受け渡しを行うために使われる。カーネル内のデータの流れ・データの状態をモニタリングすることは異常検知を行うための 1 つの手段として用いられると考えられる。

本研究では、我々が開発を行っている静的型付けを

持つスクリプト言語 Konoha<sup>1)</sup> を用いてカーネル空間で動くスクリプト言語を実装し、カーネル空間でスクリプト言語を利用することの有用性を追求している。<sup>2)</sup> これまでに Linux Kernel 上に言語処理系を Loadable Kernel Module (LKM) としてカーネル空間内に移植した。

我々は、スクリプト言語を用いたカーネル空間における利用例としてカーネルが扱う資源のモニタリングが有用であると考え、なぜなら、スクリプト言語を用いると、プログラムを再コンパイルせずに、モニターにて取得するデータを実行時に変更したり、モニターを削除したりすることができるため、実行時のカーネルを止めることなく調査を続けることができるからである。

本稿では、Konoha スクリプトを用いてカーネル空間に存在する資源に対するモニターを記述し、スクリプトによるモニタの柔軟な記述と、低コストなモニタリングを実現する。本稿ではその第一歩としてスクリプトがカーネルと連携をとるためにはメモリ上に展開されているデータ構造へのアクセス方法の提供を行う必要がある。本稿では、カーネルモニタリングを行う機構の Konoha 言語における実装状況と、そこから得

<sup>†</sup> 横浜国立大学大学院  
Graduate School of Yokohama National University

<sup>☆</sup> 現在、日本科学技術振興機構 CREST  
Presently with Japan Science and Technology Agency/CREST

た知見について述べる。

## 2. システム構成

実装している言語システムの全体図を図1に示す。本システムではまず、Cで記述されたKonoha言語処理系をカーネル空間で動作させ、ユーザ空間から受け取ったスクリプトをカーネル空間上でバイトコードに変換する。次に、Konohaのオブジェクトとカーネル内の資源と対応関係をC言語で記述し、この対応表を用いてカーネル内の資源を操作し、スクリプトの実行を行う。

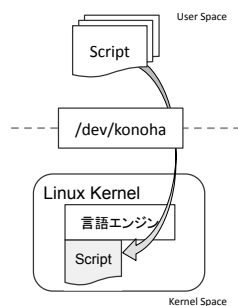


図1 Konoha言語処理系の実行の流れ

## 3. カーネル内データのトレース

本システムでモニタリング対象は、カーネル内で定義されている変数から、辿ることのできるデータのすべてを対象としている。スクリプト上で指定した周期毎にデータを取得し、読み取った値をログとして保存する形を取る。スクリプト上でメモリ空間上に配置されているバイト列からカーネルのデータ構造体の情報を推論し、モニタリングを行うことは現実的ではない。我々は、スクリプト上からカーネルのデータ構造体にアクセスを行うために、コンパイル時に利用したカーネルのソースコードから構造体メンバの情報を収集し、スクリプトとカーネルのデータ構造を連携する仕組みを導入し、実現をしている。

本システムにおけるカーネル情報の取得は以下の手順で行われる。まずカーネルのソースコードをCコンパイラの1つであるClangを用いて一旦ASTに変換し、カーネル内で利用されているデータ構造の情報を収集する。

次に、モニタスクリプトを記述し、スクリプトが正しく動作するかを検証するため、一旦ユーザ空間上で

スクリプトのコンパイルを行う。この時、収集されたデータ構造情報とカーネル上に展開されているデータのアドレス情報から、モニタリングに用いるデータを取得するのに必要なアドレス情報、構造体メンバのオフセットを計算し、カーネルモジュールとしてコンパイルを行う。最後に、モニタスクリプトをカーネル空間に転送し、すでにカーネルモジュールとして展開されているKonoha言語処理系でコンパイルを行い、スクリプト実行を行う。

## 4. おわりに

本稿では、スクリプト言語Konohaをカーネル内モニタリングツールとして利用するためのカーネル内で利用されているデータ構造をトレースするための仕組みについて検討を行った。今後はカーネル空間で実験を行い、パフォーマンス評価を行う予定である。

**謝辞** 本研究は、JST/CREST「実用化を目指した組込みシステム用ディベントラブル・オペレーティングシステム」領域の研究課題「実行時の安全性を確保するSecurityWeaverとP-SCRIPT」の一部として行われた。

## 参考文献

- 1) Kuramitsu, K.: Konoha - Implementing a Static Scripting Language with Dynamic Behaviors, *Workshop on Self-sustaining Systems 2010*, S3, The University of Tokyo, Japan (2010).
- 2) 井出真広, 中田晋平, 倉光君郎: スクリプティング言語によるカーネル拡張, 情報処理学会研究報告. [システムソフトウェアとオペレーティング・システム], Vol.2010, No.6, pp. 1-6 (2010-01-20).