耐障害性が高くマルチコア性能を最大限発揮できるElixir(エリクサー)を学んでみよう

講師: 山崎 進 (北九州市立大学), 高瀬 英希 (京都大学)

長らく組込みソフトウェア開発にはC言語が使われてきました。最近ではC++やmruby, MicroPython も使われてきだしています。とくにC言語やC++には不満を覚えている人も多くいらっしゃるのではないでしょうか。また，例外処理やマルチコア対応に苦労していらっしゃる方も少なくないでしょう。

Elixir(エリクサー)は，最近登場した新しいプログラミング言語です。Elixirは次のような特徴を持っています。

1. Elixir は関数型言語です。変数は一度値が決まったらその後不変です。この性質から並列処理が得意中の得意です。私たち fukuoka.ex チームが開発したベンチマークテストでは，コア数のとても多いPC上で，他の言語と比べて高い性能を発揮することが明らかになりました。したがって，マルチコア性能を最大限に発揮できる期待が持てます。ちなみに，この特性を生かして，GPUを駆動することにも成功しています。
2. Elixir は，従来の関数型言語と違って，とてもシンプルでとっつきやすい，それでいてパワフルな言語仕様です。たとえば，パイプライン演算子とMapReduceモデルに基づいてデータを加工しながら計算が進行するように記述できます。習得が容易で，かつ記述性に優れ，結果として生産性がとても向上します。
3. Elixirで実装されたウェブサイト構築用フレームワーク Phoenix は，Ruby で実装された同様のフレームワークである Ruby on Rails と同等以上の生産性を誇りながら，レスポンス性が極めて高いです。データベースをもとに複雑な計算をしてグラフィカルな表示をするようなウェブサイトを構築したときに，大量のアクセスがあっても耐えられます。このことにより，IoTのバックエンドサーバーを構築するときに Phoenix を用いると，極めてレスポンス性の高いシステムを構築できます。もしデバイス側でも用いることができれば，リアルタイム性に優れたシステムを構築できる潜在能力があります。
4. 1つ1つのタスク(プロセス・タスク)ごとに堅牢なメモリ管理が行き届いています。加えてレスポンス性が高く，再起動をとても高速に行えます。そのため，Elixir や Phoenix では，try / catch のような例外処理をするのではなく，障害が起こったら再起動し，外部で動作している障害監視プロセスで例外処理を行う，というようなシステム構成にすることができます。これにより，例外処理記述がとてもシンプルになり，耐障害性の高いシステムを構築できます。この性質は IoT バックエンドサーバーにとって，とてもありがたい性質です。さらに，もしデバイス側でも用いることができれば，耐障害性が高く，かつ例外処理記述をシンプルにできる可能性が高いです。

Elixir は Linux が動作する IoT ボードでは動作させて，Groveモジュールの動作に成功した実績があります。しかし，残念ながら Elixir は現在ではより省メモリのシステムに搭載したり，リアルタイム性を記述するための言語仕様を備えていたりはしていません。したがって，IoTボードや IoT バックエンドサーバーに適用する場合を除き，今すぐ組込みソフトウェア開発に応用できると

いうものではありません。しかしながら，私たちは現在，ZEAM という言語処理系を開発しています。ZEAM の狙いの1つは，Elixir で組込みシステムを開発できるようにしようとしています。

本分科会では，Elixir がどんなものかに触れ，どのような期待や展望があるのかを共有した後，ZEAM のロードマップを説明します。そのあとで習得のしかたをご紹介し，オンラインでの Elixir コミュニティをご案内します。

今までに情報処理学会で発表したプレゼンテーションのタイトル部分をご紹介します。