

IoTフレームワークで組み込み開発に挑戦

IoT 向きモダン言語

エリクサー

Elixirの研究

新連載

第1回 IoT向きプログラミング言語Elixirの世界



高瀬 英希

もはやC/C++だけでIoTや組み込みのデバイスを開発するのは大変だと考えている方は多いのではないのでしょうか。本連載では、次世代のIoT開発手段として関数型言語Elixirと、Elixirを使ったIoTフレームワークNervesを紹介します。

IoT向きモダン・プログラミング言語Elixirとは？

エリクサー

Elixirは動的型付け・関数型に分類されるプログラミング言語です⁽¹⁾。スケーラブルで保守性の高いアプリケーションを構築するために設計されました。

並行なマルチスレッド処理が得意で軽量ウェブ・サーバなどに使われるErlangというプログラミング言語があり、ElixirはこのErlangの流れをくんでいます。

開発者はJosé Valimで、

2011年からの開発開始とされており、2012年に早期版が公開されました。そして2014年9月に最初の公式版であるv1.0.0が公開され、半年に1回のペースでマイナーバージョンがリリースされています。本記事執筆時点での最新版は2020年1月27日に公開されたv1.10です⁽²⁾。ライセンスは

注1：Erlangにおける「プロセス」は、Erlang VM上のプロセスを意味します。LinuxなどのいわゆるOSのプロセスとは全く別の概念になります。

注2：全ての「もの」を「アクター」という構成要素で表現する並行計算モデル。アクター間でメッセージを受送信しながら並行かつ非同同期に処理が進行するような概念のモデルです。

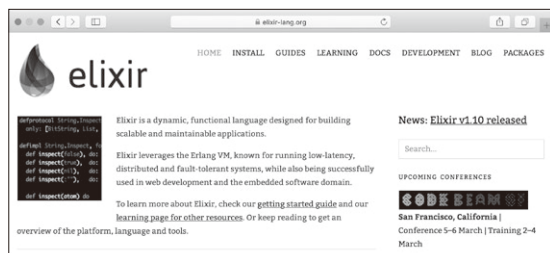


図1 多接続ウェブ・サービス処理が得意なモダン・プログラミング言語Elixir

<https://elixir-lang.org/>

Apache 2.0を採用しています(図1)。

Elixirのことを紹介するとき、筆者は「Erlangを父親として、Rubyを母親として生まれたプログラミング言語である」という表現をよく使っています。それぞれのメリットを交えながら、Elixirの特徴を紹介します。

● 並行・分散・軽量・低遅延

ElixirはErlang VM (BEAM) 上で動作します。Elixirのプログラム・コードは、Erlang VMのバイト・コードにコンパイルされて実行されます。Erlangはスウェーデンの通信機器メーカーであるエリクソンによって開発が始まり、30年以上の歴史を誇っている言語・実行環境です。もともとは電話交換機のために開発されたことから、ソフト・リアルタイムな並行処理志向の分散システム構築に適しており、低遅延性と耐障害性、高可用性といった特徴があります。

ElixirはErlangのメリットをそのまま享受できます。特にElixir/Erlangのプロセス・モデル^{注1}は、アクター・モデル^{注2}を採用していて並行処理に優れています。全てのコードはプロセスの中で動き、プロセス間のメッセージ・パッシングによって処理が進行します。また、Elixir/Erlangプロセスは、OSで生成されるプロセスと比べてメモリ使用量が少なく、かつ起動時間が短くなります。プロセスごとに独立してメモリが確保され、プロセスごとにGC(ガベージ・コレクション)が走ります。GCによってシステム全体が停止してしまうことはありません。プロセスの障害時にはプロセス単位で復旧・再起動が行えるため、堅牢性にも優れています。Elixir/Erlangで構築された分散システムでは、同時に数万個以上のプロセスを動作させながら並列性能を発揮させることも楽勝です⁽³⁾。

なおElixirとErlangは、アプリケーション内で相互に呼び出すことができます。また、Elixirには並列プログラミングを実行するためのさまざまなライブラリが用意されています。Erlangのライブラリ集Erlang/OTPのプロセス並行実行モデルをそのまま利用できるTask、AgentやGenServer、Elixir固有のGenStageやFlow、Broadwayなどがあります。