

Emscripten を用いた Web ベースパーサーとエディターの開発

Development of a Web-based editor with a parser using Emscripten

鳥原 悠平^{*1}, 香川 考司^{*2}
Yuhei TORIHARA^{*1}, Koji KAGAWA^{*2}

^{*1}香川大学大学院工学研究科

^{*1}Graduate School of Engineering, Kagawa University

^{*2}香川大学工学部

^{*2}Faculty of Engineering, Kagawa University

Email: s14g473@stmail.eng.kagawa-u.ac.jp

あらまし: プログラミング初心者がソースコードを記述する際には、文法上の誤りや変数名の誤入力などが頻繁に起こり得る。そこで、それらのエラーの解決の手助けをすることで、プログラミング学習支援を行う。その方法として、Web ベースで機能するプログラミングエディターとパーサーを提案し、開発を行った。システムの実装を行うにあたって、システムのベースとして CodeMirror を使い、そこに Emscripten を用いたパーサーや jsTree を追加することで、構文解析や構文木の生成といった機能を実装した。

キーワード: プログラミング学習, Emscripten, Web ベース

1. はじめに

プログラミング初心者がプログラミング学習を行う場合、様々な問題に直面することが想定される。例えば、プログラミング言語の文法上の誤りや変数名の間違いなどの入力ミスや、コンパイルエラー発生時の修正個所の判別が難しいことなどが挙げられる。したがって、それらの問題の解決を支援するためのシステムの開発が必要であると考えられる。入力ミスを減少させる手段としては、Eclipse のような高機能なエディターを利用することが考えられる。しかし、これらのエディターは基本的にユーザー側にインストールやバージョンアップへの対応などの作業を必要とする。よってそれらの手間を省くために学習用として Web ベースのエディターが望まれる。しかし Web ベースエディターは JavaScript で実装されるため、通常 Bison などで記述される構文解析部の実装が難しいという問題がある。

本研究では、C 言語のソースコードを JavaScript に変換できる、Emscripten⁽¹⁾ を用いた Web ベースパーサーとエディターの開発を行う。

2. システムの要件

プログラミング初心者への学習支援を行うため、学習者がプログラミングへの苦手意識を持ちにくくする必要がある。そのため、初心者の陥りがちな構文エラーの発生原因がわからないという問題を取り除く必要がある。また、入門者でも導入が容易に行えるように、Web ベースシステムとして開発する。そして、教員側から容易に拡張を行えるよう、拡張しやすいシステムでなければならない。Adobe Flex と ActionScript を用いるシステム⁽²⁾も Web ベースだが、開発環境のセットアップとコンパイルが必要なため、HTML と JavaScript に基づくシステムが、より柔軟性・拡張性が高いと考えられる。

3. Emscripten

Emscripten とは、LLVM インフラストラクチャー⁽³⁾が生成した仮想マシンの機械語 (LLVM-IR) を JavaScript のコードに変換するシステムである。これにより、LLVM 用のコンパイラが提供されている様々なプログラミング言語から JavaScript へプログラムを変換することが可能となっている。C/C++ 言語の場合の変換の流れとしてはフロントエンドである Clang を利用して C/C++ のコードを中間言語へ変換、Emscripten でこれを JavaScript に変換するというものである。

本システムでは、Web ベースのエディター上に記述された C 言語のソースコードに対して構文解析を行い、解析結果をエディター上に反映することを想定している。しかし、Web ベースのエディターは JavaScript によって構成されており、また、構文解析器は、Bison から C 言語のコードで出力されているため、そのままではエディター上で実行することはできない。そこで、LLVM コードを JavaScript に変換することが出来る Emscripten を利用する必要がある。

4. システムの概要

本研究のシステムは、ブラウザ上のエディター領域に入力された C 言語のソースコードに対して構文解析を行い、結果をブラウザ上に出力するというものである。それに加えて、解析結果をもとに構文木を作成し、これもブラウザ上に出力する。構文解析器は LLVM コードで出力されているため、それをブラウザ上で実行するために、Emscripten を利用している。また、解析結果から構文木を作成するために、解析結果を JSON で書き出しそれを jsTree という JavaScript ライブラリーを使用し構文木として出力している。学習者は、このシステムへ Web ブラウザーから通常の Web ページと同様にア

クセスし、システムを利用する。アクセスしたシステムのページ上のエディター領域に C 言語のソースコードを入力し、構文解析の開始を表すボタンを押すことで、ソースコードの構文解析を行い、その結果がブラウザ上に表示される。そして、その結果をもとに、学習者は構文エラーや入力ミスなどのエラーチェックを行い、スムーズにプログラミング学習を行える。さらに、構文木を見ることで、関数の動作やプログラムの流れについての理解を深めることが可能である。本システムの構成については、図 1 に示すとおりである。

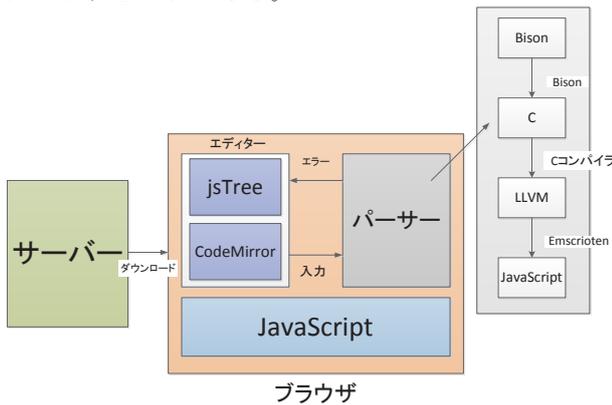


図 1 システムの構成

5. システムの実装

本研究では、CodeMirror に手を加える形で、システムの実装を行っている。そのため、CodeMirror 本来の機能である関数や変数の強調表現についてはそのまま利用している。それに加えて、構文解析器と字句解析器からなるパーサーを実装し、ソースコード上の構文エラーを検知することが出来る。さらに、jsTree を用いて、構文木を生成する機能を「式」に対してのみではあるが実装している。

今回実装できたシステムの全体図を図 2 に示している。

C Parser Demo

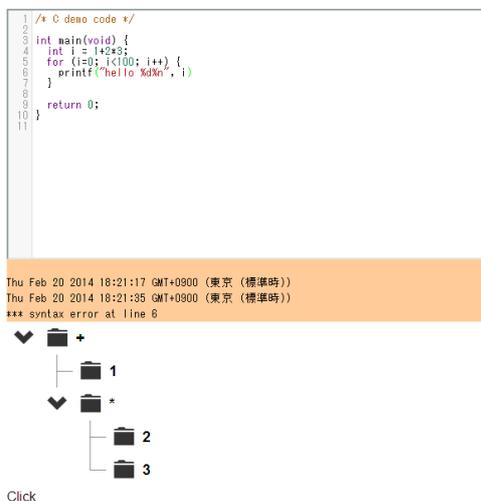


図 2 実装したシステム

6. まとめ

本研究では、プログラミング初心者へのプログラミング学習支援を行うために、Emscripten を利用して Web ベースで機能するプログラミング用のエディターとパーサーを開発した。Emscripten は C 言語を JavaScript に変換することが可能なので、そのままでは Web ブラウザー上に表示することのできない構文解析結果を、Web ブラウザー上に表示することが出来る。Web ベースのシステムであり、導入が容易であるため、学習者は学習環境づくりを気軽に行える。また、入力にミスがある場合に指摘があるため、プログラミング時の入力ミスに気づきやすいといえる。さらに、システムは JavaScript によって記述されているため、カスタマイズを行うことが出来る。これらの特徴から、1 章で述べたシステムに求められる要件を概ね満たしているといえる。しかし、構文エラーの原因の判別を行う機能はまだ未実装である。また、C 言語以外のプログラミング言語への対応など、システムの発展の余地はまだ残っているといえる。

7. 今後の課題

今回の研究では、Web ベースのエディターと C 言語パーサーの基礎の部分の実装は出来上がった。しかし、実際にプログラミングを行う際に学習者への手助けとなる構文エラーの検出や原因の判別、ユーザーの定義した変数や関数への対応、入力ミスがあった場合の指摘などの機能は未実装である。また、プログラミング言語の文法や構文への理解を深めるために、入力したソースコードをもとに C プログラム全体に対する構文木を生成し、表示する機能を追加する必要があるといえる。プログラム全体に対する構文解析は時間がかかる可能性があるが、WebWorker を利用することで、構文解析をバックグラウンドで行うことが出来るようになる。

参考文献

- (1) Alon Zakai, “Emscripten: an LLVM-to-JavaScript compiler”, SPLASH '11, pp.301-312, (2011)
- (2) 中山・香川, “AMF を利用したプログラミング学習支援システム” JSiSE 研究報告 25 (3) pp.7-12 (2010)
- (3) Chris Lattner, Vikram Adve, “LLVM: a compilation framework for lifelong program analysis & transformation”, CGO 2004, pp.75-86 (2004)