

JavaScript 上で動作する C 言語用視覚的デバッガの開発

Development of a visual debugger for C that runs on JavaScript

長江明彦^{*1}, 香川考司^{*2}
Akihiko NAGAE^{*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: s12g472@stmail.eng.kagawa-u.ac.jp

あらまし: プログラミング初心学習者にとって、意図しない挙動をとるプログラムの修正は難しい。デバッガの利用が挙げられるが、導入作業や使い方の学習は障害となる。本研究では、C 言語を対象とした Web ブラウザ上で動作する視覚的なデバッガを提案する。システムは HTML + CSS + JavaScript で構成され、教員によるカスタマイズが可能なものとし、各実行情報に応じたアニメーションなどの情報を提供する。また、C 言語は一般的に機械語にコンパイルして実行するため、Web ブラウザ上での実行は難しい。そこで、本研究では C 言語のインタプリタを JavaScript で実装する。

キーワード: C 言語, プログラミング教育, Web ベース, デバッガ, インタプリタ

1. はじめに

プログラミング初心学習者の多くは、プログラムが裏でどのように動いているかの理解が不十分である。このため、期待と異なる実行結果に陥った場合、修正することが困難である。これを補助するツールとしてデバッガが提案されるが、初心学習者にとって、いきなりデバッガを利用することは難しい。導入が大変であったり、機能が多すぎたり、デバッガを使う上で新たに学ばなければならない知識があったりと、問題を抱えている。また、多くのデバッガは、ブレークポイントまで一気に実行するか、文単位でのステップ実行のみである。このため、学習者の理解度に合わせた情報の提供が十分に行えていない。これらの問題を解決することで、初心学習者でもデバッガを利用できると考えている。

そこで、本研究では導入が不要で、教員によりカスタマイズが可能な、Web ベースのデバッガを開発した。学習者が入力したソースコードを元に、プログラムをステップ実行する。システムの利用対象者は C 言語初心学習者とし、初心学習者があまり扱わない仕様を取り除いた、C 言語のサブセット用のデバッガを実装した。これは、筆者らの所属学科で最初に C 言語を学習しているためである。また、C 言語を Web ブラウザ上で動作させる環境は少ない。このため、本研究ではこれを実装した。

2. XCI の利用

本研究のシステムでは、XCI⁽¹⁾を利用する。XCI とは C 言語で書かれた C 言語のインタプリタであり、XML 形式で構文木を出力可能である。XCI を用いた Web ベースの処理系は既に存在する⁽²⁾が、後述するデバッガの条件を満たしていない。XCI の解釈部事態を改造することは難しいため、XCI の構文解析

部までを利用することにした。構文木を元にステップ実行することで、ソースコードに対応したプログラムの実行を実現した。

3. C 言語のサブセット

本システムで扱う言語は、C 言語のサブセットである。これは、初心学習者があまり使わない仕様を C 言語から除去したものである。具体的には、ポインタ、コマンドライン引数、構造体および共用体、列挙体、typedef、静的変数、その他入門的でない型修飾子を除去した。

また、本システムは意図しない挙動をとるプログラムの修正を目的としている。そのため、未初期化の変数になんらかの値を準備したり、配列の範囲外へアクセスしたりといったことは行わない。そのような場合、エラーとして学習者に通知する。

4. システム構成

本システムは、Web ブラウザ上で実行できるデバッガである。本システムのページにアクセスし、本システムが対応している C 言語のサブセットのソースコードを入力することで、プログラムのステップ実行を行うことが可能である。

図 1 に示す構成となっており、大きく分けて、学習者のソースコードを解析するバックエンド、プログラムをステップ実行するインタプリタ、UI の提供とインタプリタの制御を行うフロントエンドの 3 つのコンポーネントから成る。バックエンドはサーバ、フロントエンドはクライアント上で動作する。また、インタプリタはフロントエンドの一部となる。

なお、現在は構文解析をサーバ上で行っているが、今後解析をクライアント上で行うようにすることを検討している。

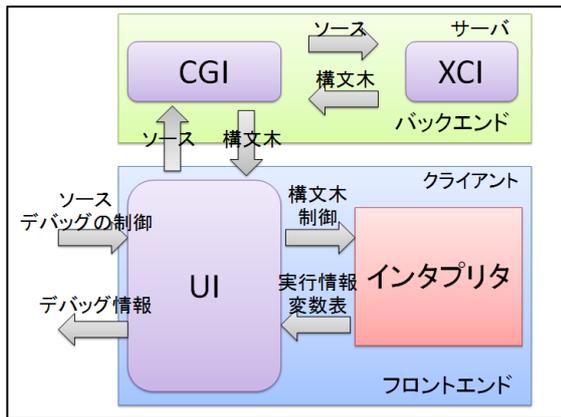


図1 システムの構成

5. バックエンド

バックエンドは、受信したソースコードからJSON形式の構文木を生成する。XCIを起動し、受信したソースコードから構文木を生成する。この構文木を元に、本システムで利用するJSON形式の構文木を生成する。対応していない仕様の除去などを行った後、フロントエンドに生成した構文木を送る。

6. インタプリタ

デバッガは、ステップ実行という形でプログラムを細かく分けて実行する。このため、一度実行を開始したら終了するまで止まらない処理系は、デバッガとしては不適である。プログラムを何らかの単位で分割し、その単位ごとに実行する必要がある。機械語への変換することで、各命令を実行の単位としてプログラムの実行を停止および継続することが可能である。しかし、この場合ソースコードと実行との対応が失われてしまう。

本システムではプログラムの実行単位を構文木の要素にした。構文木は、プログラムの論理的な構造そのものである。これに基づいて実行すれば、実行とソースコードとの対応を維持することが出来る。また、構文木は木構造であり、多くの要素が集まって出来ている。この要素1つ1つを実行の単位とし、木構造の要素をたどることで、ソースコードに基づいたデバッグ実行を行うことが可能となる。

本システムでは、構文木を元にステップ実行を行うインタプリタを、Webブラウザ上で動作させるためにJavaScriptによって実装した。ステップ実行ごとに、実行結果情報を返す。

7. フロントエンド

フロントエンドは、クライアント上で動作する部分である。図2に示すようにHTMLなどで構成されたWebページである。ユーザからのソースコードや、プログラムのステップ実行の制御の入力を担う。

また、インタプリタの制御を行う。インタプリタが行うステップ実行の実行結果情報を元に、処理の継続や停止、そしてUIや画面表示を更新する。教員がカスタマイズする部分であり、このカスタマイ

ズによって学習者の理解度に応じた情報および機能の提供を行う。例えば、「if文における式の評価」に着目するのであれば、該当する要素の実行ごとにインタプリタの処理を停止し、その他の場合は連続して処理を行う。そして、処理を停止した時点での要素に対応するソースコードをハイライトすることで、該当箇所を強調することができる。

加え、実行結果情報を元に、対応するアニメーションを画面に表示する機能を提供する。計算や代入などをアニメーションで表現することにより、学習者の理解を補助する。さらに、インタプリタが保持する変数表は、外部からアクセス可能である。これを元に、実際に画面上に変数一覧を表示や、実行時の変数値変更を提供する。

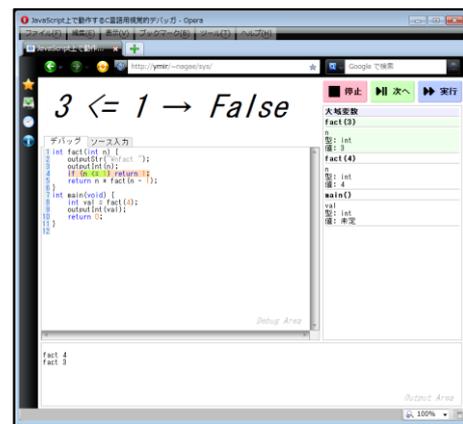


図2 システムのスクリーンショット

8. おわりに

C言語の初心学習者支援を目的に、Webベースのデバッガを開発した。導入の作業を行う必要がなく、教員によるカスタマイズにより学習者の理解度に応じた情報や機能の提供ができる。XCIが生成した構文木を元に、JavaScriptで実装したインタプリタでプログラムのステップ実行を行う。フロントエンドにおけるインタプリタの制御箇所を教員がカスタマイズすることで、学習者のコードにおける特定の箇所の強調が可能となる。また、アニメーションにより情報を提供する欄があり、これも教員によるカスタマイズが可能である。

謝辞

本研究は科研費(23501152)の助成を受けたものである。

参考文献

- (1) 権藤克彦, 川島勇人: “コンパクトなANSI CインタプリタXCIの設計と実装”, 電子情報通信学会論文誌. D-I, 情報・システム, I-情報処理, pp159-168 (2003)
- (2) 東山裕徳, 香川考司: “Adobe Alchemyを用いたWeb上の言語処理系”, 情報処理学会第103回コンピュータと教育研究発表会 (CE103) (2010)